

Recibido 15 Jul 2021

ReCIBE, Año 10 No. 1, Mayo 2021

Aceptado 02 Sep 2021

Prácticas de los equipos de desarrollo de microservicios: un mapeo sistemático de la literatura

Microservices team practices: a systematic mapping study

Alan González-Heredia¹
alan14_04@hotmail.com

Jorge Octavio Ocharán-Hernández¹
jocharan@uv.mx

Maria de los Ángeles Arenas-Valdés¹
aarenas@uv.mx

Karen Cortés Verdín¹
kcortes@uv.mx

¹Facultad de Estadística e Informática, Universidad Veracruzana, México

RESUMEN

Debido a la creciente adopción de la Arquitectura de Microservicios, estilo que hace énfasis en la división de los sistemas en una colección de pequeños servicios con una sola responsabilidad, existe una necesidad de construir este tipo de sistemas con calidad y de adaptar las prácticas de los equipos de desarrollo para alcanzar una mejor escalabilidad, mantenibilidad, facilidad de despliegue así como una mayor agilidad y una clara separación de intereses. Al igual que con otro tipo de sistemas, el desarrollo de microservicios trae consigo diferentes retos que las organizaciones deben enfrentar. Dichos retos no solo tienen que ver con la parte técnica, sino que también implica un cambio en las prácticas y organización de los equipos de desarrollo. El objetivo de este estudio es analizar las prácticas de desarrollo que llevan a cabo los equipos, la forma en que éstos se organizan y los retos que enfrentan durante el desarrollo de microservicios. Siguiendo un proceso de mapeo sistemático de la literatura se recopilaron 26 estudios primarios los cuales fueron analizados mediante una síntesis temática. Los resultados muestran que los equipos tienden a ser pequeños y multifuncionales, las prácticas que realizan están relacionadas con una cultura ágil y de independencia, lo que se complementa con los modelos de gobernanza descentralizada y ciclo de vida independiente. Los retos se relacionan con la complejidad del dominio, aspectos específicos de la organización y las habilidades que los desarrolladores deben de tener. Como conclusión se obtuvo la importancia de las características de una organización y sus equipos de desarrollo, la estrecha relación que tiene una cultura DevOps con el desarrollo de microservicios, la independencia y descomposición son aspectos clave que deben tomarse en cuenta y se detectaron diversos retos donde el factor humano juega un papel importante.

PALABRAS CLAVE

Arquitectura de microservicios, Prácticas de desarrollo, Mapeo sistemático, Síntesis temática

ABSTRACT

Due to the growing adoption of Microservices Architecture, a style that emphasizes the division of systems into a collection of small services with a single responsibility, there is a need to build this type of system with quality and to adapt the practices of development teams to achieve better scalability, maintainability, ease of deployment as well as greater agility and a clear separation of interests. As with other types of systems, the development of microservices brings with it different challenges that organizations must face. These challenges are related to the technical side and imply a change in the practices and organization of the development teams. This study aims to analyze the development practices carried out by the teams, the way they are organized, and the challenges they face during the development of microservices. Following a systematic literature mapping process, 26 primary studies were collected and analyzed through a thematic synthesis. The results show that teams tend to be small and cross-functional. Their practices are related to an agile and independent culture, complemented by decentralized governance and independent lifecycle models. The challenges are related to the complexity of the domain, specific aspects of the organization, and the skills that developers must-have. In conclusion, the importance of the characteristics of an organization and its development teams, the close relationship that a DevOps culture has with the development of microservices, independence, and decomposition are vital aspects that must be taken into account, and several challenges were detected where the human factor plays an important role.

KEYWORDS

Microservice Architecture, Development Practices, Systematic Mapping, Thematic Synthesis

1. INTRODUCCIÓN

En los últimos años la arquitectura de microservicios se ha convertido en una opción de estilo arquitectónico dominante en la industria del software orientado a servicios. Las empresas han descubierto que adoptar un enfoque de microservicios permite que los equipos y las organizaciones de software sean más productivos en general y desarrollen productos de software exitosos. Empresas como Netflix, SoundCloud, Amazon, Uber han adoptado dicha arquitectura y obtenido sus beneficios. De acuerdo con Larrucea (2018) los microservicios están inspirados en la computación orientada a servicios (SOA); son aplicaciones pequeñas con una sola responsabilidad, que se pueden implementar, escalar y probar de forma independiente. Sin embargo, al adoptar o querer migrar a una arquitectura de microservicios hay factores organizacionales que deben tomarse en cuenta. Estos factores tienen que ver con las actividades que se realizan durante el desarrollo y las características que poseen los equipos. Los microservicios tienen el fin de crear un sistema modular esto trae consigo las metodologías ágiles que permiten a los equipos trabajar de manera independiente. Uno de los problemas que pueden generarse es que los equipos estén cada vez más aislados unos de otros y tengan una poca o nula comunicación, lo que generará problemas en el desarrollo del sistema entero, ya que, al igual que los microservicios, los equipos deben trabajar juntos para conseguir un objetivo (Fowler, 2016). En 1968 Melvin E. Conway realizó un estudio en el cual determinó que “las organizaciones que diseñan sistemas están limitadas a producir diseños que son copias de las estructuras de comunicación de dichas organizaciones” (Conway, 1968). También existe evidencia de distintas organizaciones que migraron de monolito a microservicios, se detectó que uno de los desafíos más importantes eran los organizacionales debido a que los equipos tuvieron que hacer frente a un cambio de mentalidad al tener que establecer procesos ágiles (Fritzsche et al., 2019). De igual forma Salah (2016) reportaron que la capacitación de personal es una tarea que consume mucho tiempo y si a eso se le añade la flexibilidad que promueve la arquitectura en cuanto a las tecnologías de desarrollo es posible que la coordinación entre diferentes equipos se vuelva compleja. El objetivo de este estudio de mapeo sistemático es analizar el estado actual en el que se encuentran las organizaciones que desarrollan sistemas con arquitectura de microservicios y las prácticas que sus equipos realizan durante el proceso de desarrollo. A pesar de que se encuentran investigaciones que mencionan las prácticas de las organizaciones, no se tiene conocimiento de una revisión sistemática que reúna el estado del arte de lo que están haciendo los equipos que desarrollan con una arquitectura de microservicios, por lo cual es necesario tener esta revisión sistemática y proporcionar una fuente bibliográfica donde se recopile y analice dicha información sobre las prácticas que realizan los equipos de desarrollo.

2. TRABAJO RELACIONADO

En diversas revisiones sistemáticas enfocadas en la arquitectura de microservicios se mencionan factores importantes durante su desarrollo y los retos que conlleva. En ellas se habla sobre los aspectos organizacionales de los equipos y las prácticas que tienen que realizar para lograr la adopción de los microservicios. Pahl y Jamshidi (2016) realizaron un estudio de mapeo sistemático de 21 artículos publicados que reportaron experiencias acerca de los microservicios y su aplicación. Concluyeron que las investigaciones de microservicios aún estaban en una etapa temprana, las herramientas para facilitar su operación aún no eran suficientes y entre las futuras tendencias de investigación se encontraban, migración de microservicios, monitoreo de microservicios y DevOps con microservicios. De igual manera Di Francesco et al. (2017) llevaron a cabo un estudio de mapeo sistemático que incluyó a 71 estudios primarios, logrando una clasificación y evaluación de tendencias de publicación, enfoques de investigación y potencial de la adopción industrial para microservicios. Di Francesco et al. (2019) extendieron dicha investigación incluyendo 32 artículos más. Los autores encontraron que la investigación existente tiene un enfoque en paradigmas de la nube y que la complejidad de los microservicios se ve compensada gracias a su flexibilidad, también mencionan como las organizaciones adoptan DevOps durante el ciclo de vida de desarrollo e implica un cambio cultural, pero no profundizan en prácticas organizacionales.

Por otra parte Hassan et al. (2020) durante su estudio de mapeo sistemático sobre la transición hacia microservicios y su problema de granularidad, en sus resultados proporciona un marco de clasificación en el cual las actividades organizacionales se encuentran entre las más realizadas para una adopción satisfactoria de microservicios y por lo tanto hay que brindarles atención. También se encuentran encuestas que proporcionan evidencia de las prácticas y características de los equipos de desarrollo que están trabajando en un ambiente profesional. Zhang et al. (2019) realizaron una investigación empírica que brinda una visión sistemática de las prácticas industriales en microservicios. Guiaron su estudio con el objetivo de investigar la brecha entre las visiones ideales y las prácticas industriales reales sobre microservicios y qué beneficios se pueden obtener de las experiencias industriales. Como resultados obtuvieron una caracterización de las brechas entre las características típicas aceptadas en la comunidad y las prácticas en la industria de los microservicios, también confirmaron los beneficios que pueden obtenerse a través del desarrollo de microservicios y los posibles problemas que deben abordarse con cierta experiencia, algunos problemas descritos, por ejemplo: transformación organizacional, descomposición, distributed monitoring y localización de errores fueron hallados durante el estudio. Wang y Kadiyala (2020) realizaron un estudio amplio de métodos mixtos que incluye entrevistas en profundidad con 21 profesionales y una encuesta en línea a 37 participantes, que abarca 37 empresas en total. Concluyeron con una categorización de las prácticas en tres; arquitectura, infraestructura y administración de código, cada una de ellas está conformada por las distintas prácticas que realizan las organizaciones y también ofrecen una distribución respecto a la experiencia de los integrantes y el tamaño de los equipos. Finalmente, Fritzsche et al. (2019) investigaron el proceso de migración de 14 sistemas en diferentes dominios y tamaños mediante la realización de 16 entrevistas en profundidad con profesionales de software de 10 empresas. Entre sus descubrimientos se encuentran que, gracias a la alta complejidad de sus sistemas heredados, la mayoría de las empresas prefirieron reconstruir el sistema utilizando las tecnologías actuales en lugar de dividir las bases de código existentes, ya que no contaban con un método adecuado de descomposición. Los desafíos organizacionales estaban relacionados con las grandes y tradicionales empresas que establecían simultáneamente procesos ágiles. Iniciar un cambio de mentalidad y garantizar una colaboración fluida entre los equipos fue crucial para ellos.

3. MÉTODO DE INVESTIGACIÓN

El método de investigación seguido es el propuesto por Kitchenham et al. (2015) en su guía para realizar revisiones y mapeos sistemáticos de la literatura en el área de ingeniería de software. Dicha guía proporciona pautas que se ajustan al objetivo de las preguntas de investigación que es analizar el estado actual de la literatura, para posteriormente proponer una categorización. En esta sección se detallan las actividades que se realizaron durante la planeación y la conducción de dicho método. Se cubren los siguientes puntos: Preguntas de investigación, estrategia de búsqueda, términos de búsqueda, construcción de la cadena de búsqueda, evaluación de la cadena, proceso de selección de estudios, selección de fuentes, extracción de datos, síntesis de la información y evaluación de la calidad.

4. PLANEACIÓN

Con el propósito de explorar las iniciativas en torno a procesos de IS, en los cuales se ha utilizado la gamificación como estrategia, se establecieron un conjunto de preguntas que orientan el estudio.

4.1 Preguntas de investigación

Se especifican las siguientes preguntas de investigación (PI) con el objetivo de obtener un subconjunto o grupo de datos cualitativos durante la revisión de la literatura. Las respuestas a las preguntas de investigación proporcionan un aporte significativo a los investigadores y al público interesado en el tema de los microservicios en el ámbito organizacional, debido a que brindan información sobre el estado del arte de las organizaciones que desarrollan con arquitectura de microservicios.

PI1. ¿Qué características tienen los equipos que desarrollan sistemas con arquitectura de microservicios?

Motivación: Es de vital importancia para la investigación conocer las características de los equipos de desarrollo, debido a que la arquitectura de microservicios trae consigo cambios en ellos.

PI2. ¿Cuáles son las prácticas que realizan los equipos de organizaciones que desarrollan sistemas con arquitectura de microservicios?

Motivación: Identificar las prácticas que las organizaciones están realizando actualmente.

PI3. ¿Cómo se organizan los equipos que desarrollan sistemas con arquitectura de microservicios?

Motivación: Descubrir de que manera los equipos de desarrollo están organizándose para lograr un desarrollo con microservicios satisfactorio.

PI4: ¿Cuáles son los retos que conllevan las prácticas de los equipos en el desarrollo con arquitectura de microservicios?

Motivación: Detectar los retos que las organizaciones enfrentan durante las prácticas podrá establecer un panorama acerca de cuales son las más realizadas y como enfrentarlos anticipadamente.

4.2 Estrategia de búsqueda

Para la búsqueda de estudios se realiza una búsqueda automatizada. Primero se definen los términos de búsqueda, con base en ellos se construye la cadena de búsqueda, posteriormente para validar la completitud del proceso de búsqueda se realiza una lectura preliminar en la que se detectan revisiones de la literatura relacionadas con el objeto de estudio y se localizan los artículos que ellas incluyeron, de esta forma se obtiene un conjunto de 30 investigaciones relacionadas que son suficientes para la evaluación de la completitud en un estudio de mapeo sistemático de la literatura según sugieren Kitchenham et al. (2015) en su guía, después se realiza la evaluación de la cadena de búsqueda y finalmente se definen las fuentes a consultar.

4.3 Terminos de búsqueda

Partiendo de las preguntas de investigación y del análisis preliminar de los términos utilizados en investigaciones relacionadas, se plantean los siguientes términos para realizar la búsqueda. En la Tabla 1 se muestran los términos, los que se encuentran en la primera fila tienen el objetivo de recolectar los estudios relacionados únicamente con microservicios, los siguientes son para conocer el ámbito en que se está realizando dicha arquitectura y los últimos son para obtener las prácticas o actividades que están realizando los equipos de desarrollo en las organizaciones.

Términos	Términos relacionados
Microservice	Microservice architecture, microservice system, microservice style
Development	Organization, team, migration, implementation, industrial, agile, adoption
Practice	Experience, strategies, process, practice issue, practice challenge, scheme

Tabla 1 Términos de búsqueda

4.4 Cadena de búsqueda

Con base en los términos anteriores se realizaron prototipos de la cadena de búsqueda los cuales fueron probados en las bases de datos seleccionadas y evaluadas. Gracias a los resultados se logró determinar la cadena final, que se muestra a continuación:

(microservice* OR “microservice style” OR “microservice architecture”) AND (“development” OR “organization” OR “migration” OR “migrating” OR “implementation” OR “industrial” OR “agile” OR “adoption”) AND (practice* OR experience* OR strategy* OR process* OR challenge*)

La cadena es una combinación de los términos clave y relacionados, con el fin de lograr una búsqueda más específica se dividen en grupos para unirlos con los conectores AND y OR. La evaluación de la efectividad de la cadena fue realizada mediante los métodos recall y precision mencionados por Dieste et al. (2009).

4.5 Fuentes seleccionadas

Las fuentes de información seleccionadas se muestran en la Tabla 2. Entre ellas se encuentran bases de datos académicas cuya calidad de estudios sobre la computación es reconocida y también se han hallado estudios relacionados con el tema de microservicios.

Fuente	Dirección
IEEE Xplore	https://ieeexplore.ieee.org
ACM Digital Library	https://dl.acm.org
ScienceDirect	https://www.elsevier.com
Springer Link	https://www.springer.com
Wiley Online Library	https://onlinelibrary.wiley.com

Tabla 2 Fuentes seleccionadas

4.6 Proceso de selección

El proceso de selección de estudios primarios consiste en ingresar la cadena de búsqueda en las bases seleccionadas y filtrarlos con base en los criterios de inclusión y exclusión que se muestran en la Tabla 3. El proceso está dividido en tres fases principales que se encuentran explicadas a continuación:

Fase 1: Ingresar la cadena de búsqueda en las bases de datos seleccionadas y filtrar los estudios con base en los criterios CI-1, CI-2 y CE-1.

Fase 2: Durante esta fase al leer el título y el abstract de los trabajos serán aplicados los criterios CI-3, CI-4 y CE2.

Fase 3: En la etapa final del proceso se realizará la lectura completa de los artículos para poder determinar su selección tomando en cuenta CI-5 y CE-3.

Criterios de inclusión	Criterios de exclusión
CI-1 El artículo fue publicado entre el año 2015 y 2020	CE-1 Es un artículo de opinión, resumen o que no haya sido revisado por pares.
CI-2 Se encuentra escrito en idioma inglés	CE-2 Se centra en un enfoque de arquitectura basado en servicios, pero no en uno de microservicios.
CI-3 El título o el abstract se relaciona con algunas de las preguntas de investigación	CE-3 El artículo no proporciona detalles acerca de las prácticas mencionadas.
CI-4 El título o el abstract debe contener al menos dos términos de búsqueda	
CI-5 Responde por lo menos una de las preguntas de investigación	

Tabla 3 Criterios de selección

El intervalo de fechas definido en el CI-1 es debido a que es en dichos años cuando el tema de microservicios empezó a tener un auge. CI-2 debido al lenguaje universal. CI-3, CI-4 y CI-5 son planteados con base en las preguntas y objetivos de la investigación. CE-1 sirve para aumentar la calidad de los resultados y CE-2, CE-3 son para descartar aquellos estudios que puedan producir un sesgo en la investigación.

4.7 Extracción de datos

A continuación, se muestra en la Figura 1 la plantilla de extracción de datos. Los datos de la publicación son usados para extraer la información general de cada estudio seleccionado. Cada publicación puede tener muchos hallazgos, los cuales contienen los segmentos de texto que se relacionan con el objetivo y preguntas de investigación. Como se está realizando un estudio de mapeo sistemático la extracción de datos puede ser iterativa, ya que las tendencias importantes y la forma de categorización de los artículos solamente pueden hacerse evidentes a medida que se leen en las diversas fases del proceso. La herramienta MAXQDA es utilizada para analizar la información, apoyar la extracción de datos y en la etapa de la síntesis.

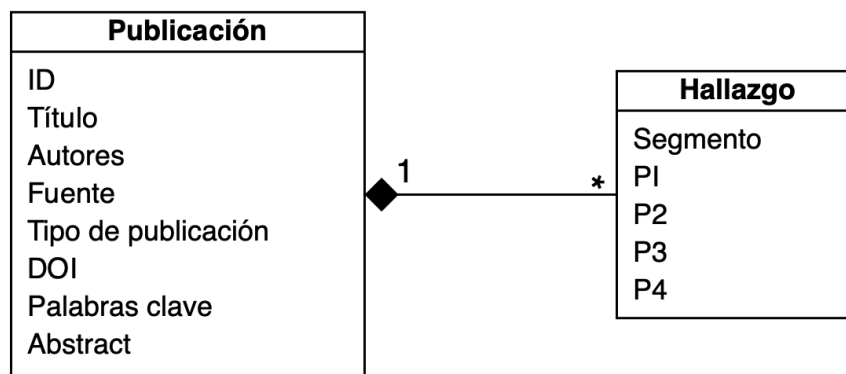


Figura 1 Plantilla de extracción de datos

4.7 Síntesis de la información

Para la síntesis de los datos se utilizó la síntesis temática siguiendo la guía propuesta por Cruzes et al. (2011). Dicho método se compone de cinco fases principales: la lectura inicial del texto, la identificación de segmentos importantes, codificación de los segmentos, la identificación de temas y temas de orden superior, como se muestra en la Figura 2. Estas actividades tienen el objetivo de hallar una clasificación a partir de la información extraída.

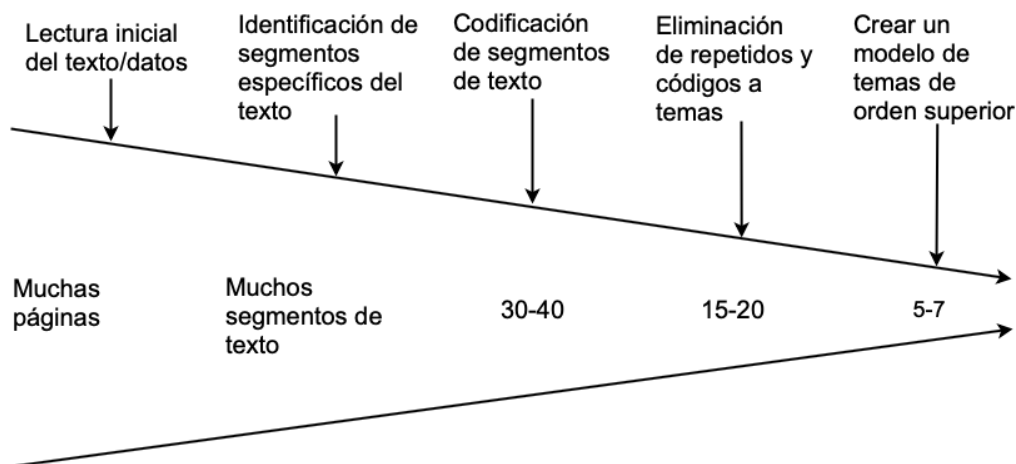


Figura 2 Proceso de la síntesis temática

5. CONDUCCIÓN

Este apartado está compuesto de dos puntos principales, el primero es la conducción del proceso de selección en el que se ejecuta la búsqueda de estudios y se aplican los criterios pertinentes en cada fase. Después, se encuentra el proceso de la síntesis temática con información detallada de cada actividad realizada durante su ejecución y la clasificación de la información obtenida.

5.1 Fase de selección

Durante la fase de selección de los estudios primero se ingresó la cadena de búsqueda en las bases de datos seleccionadas, obteniendo un resultado inicial de 2694 estudios. Posteriormente al aplicar la fase 1 quedaron un total de 1585, en esta fase se excluyeron los estudios que no estuvieran publicados entre los años 2015 y 2020, escritos en un idioma distinto al inglés y que no sean artículos de investigación o revisados por pares. Durante la fase 2 el título y abstract de los estudios fueron analizados para poder determinar su inclusión con base en la relación que tengan con las preguntas de investigación y excluirlos si no están relacionados con el tema de microservicios, el resultado al final de la fase 2 fue de 114. Finalmente, en la fase 3 se leyó completamente cada estudio restante de tal forma que se pudo determinar si respondía algunas de las preguntas de investigación y si proporcionaba información relevante sobre las prácticas durante el desarrollo de microservicios, quedando como 26 el número de estudios seleccionados para esta investigación. En la Tabla 4 se puede observar con más detalle el número de estudios seleccionados por base de datos, durante la conducción de cada una de las fases del proceso de selección.

Biblioteca digital	Inicio	Fase 1	Fase 2	Fase 3
SpringerLink	460	300	6	4
ScienceDirect	455	280	7	3
ACM Digital Library	853	501	11	4
IEEE Xplore	418	369	84	14
Wiley	408	135	6	1
Total	2694	1585	114	26

Tabla 4 Proceso de selección

5.2 Síntesis temática

Durante la conducción del análisis cualitativo de la información se realizaron las diferentes fases del proceso de la síntesis temática utilizando la herramienta MAXQDA, dicha herramienta proporciona las funciones necesarias para la codificación y extracción de los segmentos de texto. Al leer todos los estudios se lograron extraer un total de 304 segmentos de texto que proporcionan información relevante para responder las preguntas de investigación, también se hallaron 113 códigos. Posteriormente al eliminar los códigos repetidos y reducirlos a temas, restaron 95 códigos y 11 temas. En la Tabla 5 se muestran los temas de orden superior determinados y los temas relacionados.

Tema de orden superior	Tema
<i>Organization</i>	<i>Team practices</i> <i>Organizational model</i>
<i>Microservice design</i>	<i>Service granularity</i> <i>Design approaches</i> <i>Decomposition techniques</i>
<i>Developer challenges</i>	<i>Testing challenges</i> <i>Operation Challenges</i> <i>Decomposition challenges</i> <i>Skills challenges</i>
<i>Deployment and operations</i>	<i>Pipelines</i> <i>Deployment</i> <i>Monitoring and logging</i>

Tabla 5 Temas de orden superior y temas

Los temas hallados fueron analizados e interpretados para crear un modelo de temas de orden superior y la relación entre ellos, en la Figura 3 se muestra el producto final del proceso.

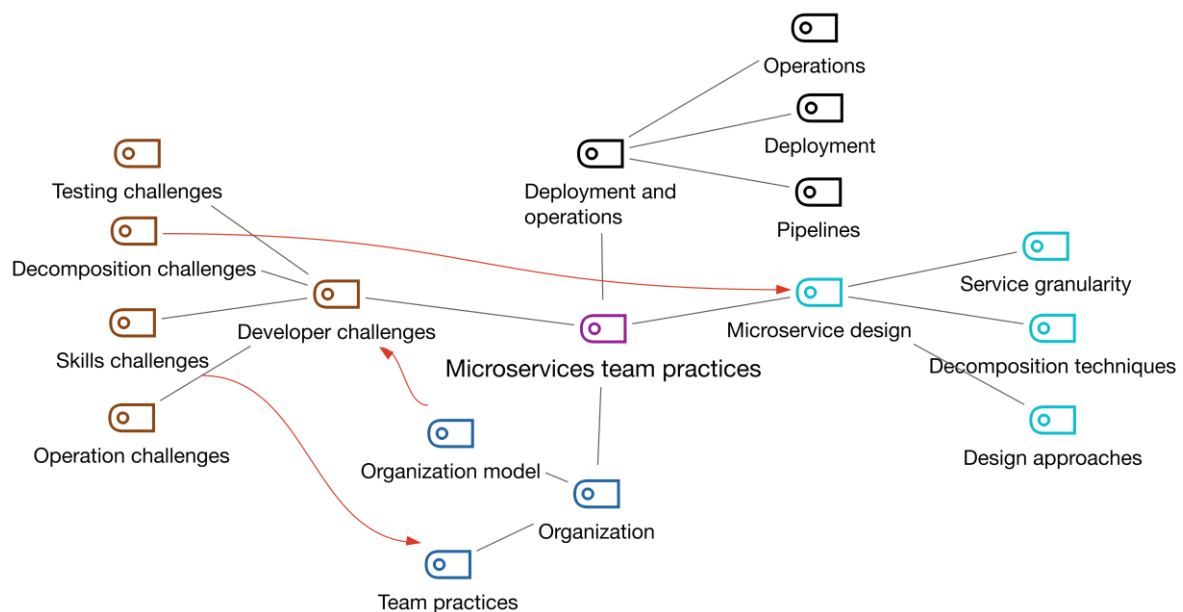


Figura 3 Modelo de orden superior

En la Tabla 6 se muestra cada uno de los temas hallados con una descripción que tiene como objetivo dar a conocer a que se refieren los códigos y segmentos de texto que se encuentran. La información completa con los segmentos de texto puede ser consultada en el siguiente enlace.

https://drive.google.com/file/d/1qMIwAYaaD_FvcHgMyzWXS-XCz6pXrQ3k/view?usp=sharing

Tema de orden superior	Tema
<i>Team practices</i>	Comprende las diversas actividades que realizan los equipos de desarrollo
<i>Organizational model</i>	Comprende los diversos modelos de organización que se adoptan durante el desarrollo
<i>Service granularity</i>	Comprende la heterogeneidad de los tamaños de los microservicios y sus límites
<i>Design approaches</i>	Comprende los diversos enfoques y patrones de diseño que son utilizados para la adopción
<i>Decomposition techniques</i>	Comprende técnicas que son usadas para descomponer un sistema o problema
<i>Testing challenges</i>	Comprende tareas que son un reto en la etapa de pruebas debido a la naturaleza de los sistemas
<i>Operation challenges</i>	Comprende los retos que enfrentan los desarrolladores durante la operación
<i>Decomposition challenges</i>	Comprende los problemas que se enfrentan durante la descomposición
<i>Skills challenges</i>	Comprende las diversas dificultades que enfrentan directamente los desarrolladores
<i>Pipelines</i>	Comprende las técnicas que son adoptadas para la automatización de pruebas y cambios
<i>Deployment</i>	Comprende las actividades que se realizan durante el despliegue de microservicios

Tabla 6 Temas obtenidos

5.1 Evaluación de la calidad

La evaluación de la calidad de los estudios primarios suele realizarse mediante un instrumento de calidad que comprende una serie de preguntas relacionadas con los objetivos, el diseño, la realización y los resultados de cada estudio. Las preguntas se denominan criterios de calidad. El instrumento de calidad suele denominarse lista de control de calidad (Kitchenham et al., 2015). El aseguramiento de la calidad de los estudios es una actividad que proporciona un valor agregado a los resultados de la investigación y ayuda a mostrar la calidad de la literatura en los estudios de mapeo sistemático. Para realizar la evaluación se utilizó una lista de verificación propuesta por Dybå y Dingsøyr (2008). Originalmente la lista está comprendida por 12 preguntas, pero se excluyeron algunas, debido al alcance de la investigación. Cada pregunta se respondió con base en la siguiente escala de valores; 0: No, 0.5: Parcialmente, 1: Sí. En la Tabla 7 se muestra un ejemplo de la lista de verificación aplicada a un estudio. Al evaluar todos los estudios el resultado promedio obtenido fue que el 100% (26) de los estudios está arriba de la puntuación de 7 y el 85% (22) tiene un puntaje total de 8. La información completa de la evaluación de calidad puede ser consultada en el siguiente enlace.

<https://drive.google.com/file/d/1aEfkZSbSRMOJHoUITcVxuun8IdLefDQW/view?usp=sharing>

ID	EP-01
<i>Is the paper based on research (or is it merely a "lessons learned" report based on expert opinion)?</i>	1
<i>Is there a clear statement of the aims of the research?</i>	1
<i>Is there an adequate description of the context in which the research was carried out?</i>	1
<i>Was the research design appropriate to address the aims of the research?</i>	1
<i>Was the data collected in a way that addressed the research issue?</i>	1
<i>Was the data analysis sufficiently rigorous?</i>	1
<i>Is there a clear statement of findings?</i>	1
<i>Is the study of value for research or practice?</i>	1
Total:	8

Tabla 7 Ejemplo de evaluación de la calidad

6. RESULTADOS

En esta sección se abordan los resultados de la conducción del método de investigación. Se extrajeron un total de 26 artículos de los cuales la cantidad de estudios en el 2016 es de tan solo 2, en cambio en el 2017 el número aumenta a 6 estudios, en el 2018 se encuentran 8 lo que es cuatro veces más que hace 2 años, para 2019 se mantiene en 7. Sin embargo, en el año 2020 se nota un decremento en el número de estudios publicados reduciéndose 3, la Figura 4 muestra la información mencionada.

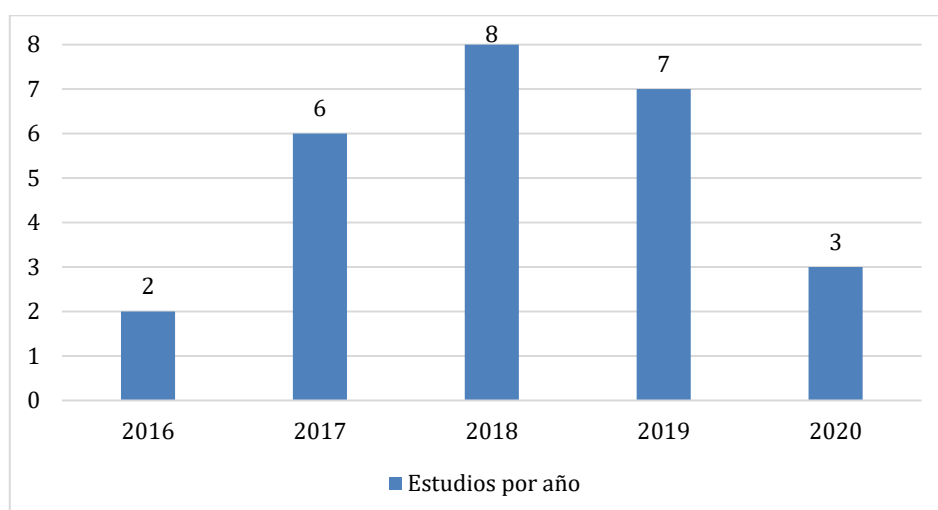


Figura 4 Estudios por año

El mayor porcentaje de los tipos de los estudios seleccionados son de conferencias con un 69% (18) del total, por otra parte están los journals con un 31% (9) de todos los recolectados, dichos valores se pueden observar en la Figura 5.

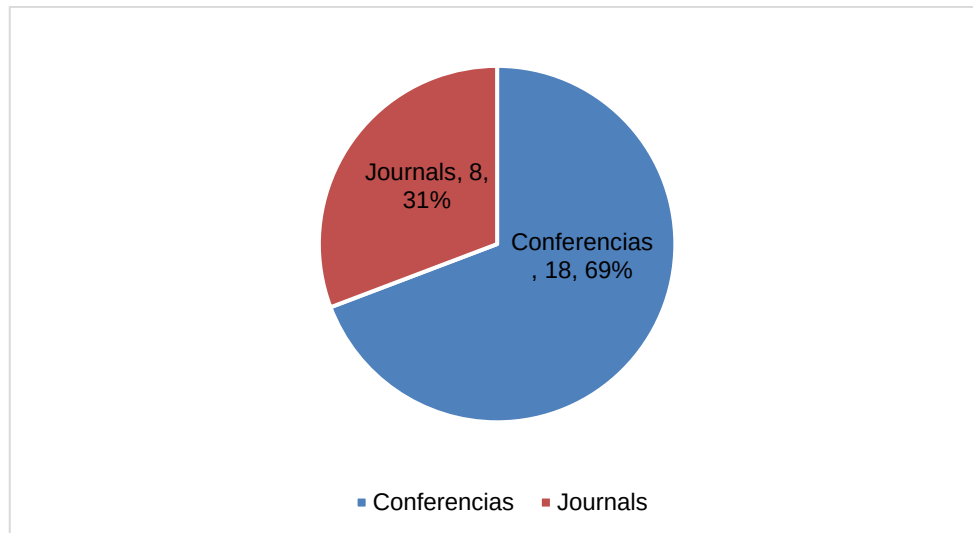


Figura 5 Tipos de estudios

En la Figura 6 se muestra que la base de datos académica donde se logró recuperar más artículos fue la IEEE Xplore con un total de 14 estudios recolectados. Seguida por ACM y Springer Link con un total de 4 estudios recolectados en cada una. Con menos estudios recuperados se encuentra Science Direct con un total de 3 y Wiley con 1 estudio.

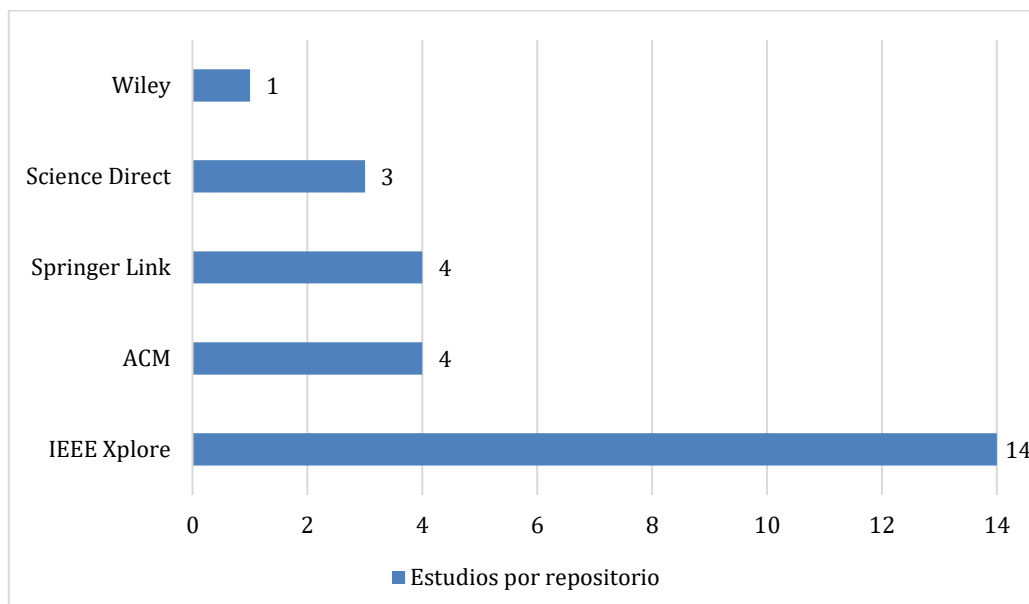


Figura 6 Estudios por repositorio

6.1 Respuestas a las preguntas de investigación

6.1.1 Características de los equipos de desarrollo

En esta sección se reportan las características que se encuentran presentes en los equipos que desarrollan sistemas con Arquitectura de Microservicios para caracterizarlos. Cabe destacar que, a pesar de que existe un número importante de trabajos que tratan el desarrollo de Microservicios, son relativamente pocos los estudios que describen las características de los equipos de desarrollo; sin embargo, es posible identificar algunos rasgos comunes. Los equipos de desarrollo tienen un tamaño de cinco a diez integrantes y tienen bajo su responsabilidad un solo servicio. En la literatura se identificó que las organizaciones siguen la regla de dos pizzas por equipo introducida en Amazon (AWS, 2020) para determinar el número de desarrolladores y que al tener asignado un solo servicio se sigue la filosofía de “you build, you run it” (Bogner et al., 2019), también mencionada en la literatura. Por otro lado, el tamaño del equipo los obliga a ser multifuncionales (Waseem et al., 2020), esto quiere decir que los integrantes deben de tener la capacidad para realizar diferentes tipos de prácticas durante todo el ciclo de vida de desarrollo. En cambio, hay estudios que reportan la existencia de equipos DevOps o de tamaño no tan pequeño como en otros casos. Dicha variación sucede ya que el dominio, alguna característica en específico o ciertas limitaciones, ocasionan que la distribución de los equipos y responsabilidades se adapte a las necesidades específicas del contexto en el que se desarrolla el sistema.

6.1.2 Prácticas de los equipos de desarrollo

Por la naturaleza de los microservicios, las organizaciones realizan prácticas complementarias que se ajustan a los dominios, pero en general son similares. Diversas prácticas mencionadas ya existían y las organizaciones han tenido que adaptarlas a este tipo de desarrollo. En la Figura 7 se muestra una gráfica con las prácticas que realizan los equipos de desarrollo por frecuencia de menciones. En primer lugar, se encuentra la selección del tipo de Application Programming Interface (API) adecuada para la comunicación entre los Microservicios (Bucchiarone et al., 2018; Fan & Ma, 2017; Lotz et al., 2019; Luz et al., 2018; Premchand & Choudhry, 2018; Salah et al., 2016; Zhang et al., 2019). Los tipos de API web puede ser RESTful, RPC o incluso message brokers para poder lograr esa sincronía entre los diversos servicios. Otras de las prácticas tiene que ver con la heterogeneidad tecnológica en la que los equipos se encuentran envueltos (Bogner et al., 2019; Garriga, 2018; Hassan et al., 2020; Luz et al., 2018; O’Connor et al., 2016; Soldani et al., 2018; Waseem et al., 2020), cuando cada equipo es responsable de un solo servicio las decisiones tecnológicas quedan bajo su control, por lo tanto, se usa una amplia variedad de tecnologías. Otros estudios reportan explícitamente diferentes prácticas ágiles (Di Francesco et al., 2018; Fan & Ma, 2017; Fritzsche et al., 2019; Idoughi et al., 2017; O’Connor et al., 2016; Srikaew & Kim, 2017) y mencionan que la Arquitectura de Microservicios se relaciona fuertemente con este tipo de prácticas, ya que eso permite la flexibilidad que se pretende alcanzar en los equipos de desarrollo. Scrum (Fritzsche et al., 2019; Sousa et al., 2020), DevOps, user stories (Idoughi et al., 2017), también son reportados en la literatura y son prácticas que se complementan con una cultura ágil. Además, se reporta un cambio cultural (O’Connor et al., 2016; Premchand & Choudhry, 2018) implica equipos pequeños (Sousa et al., 2020), equipos de operaciones (Bogner et al., 2019), equipos DevOps (Bogner et al., 2019; Zhang et al., 2019) y una autonomía en las decisiones que toman los equipos en general. La variación en la organización de los equipos se debe, de acuerdo con los autores, a los dominios y características en específico de cada uno de los sistemas. Finalmente, la alineación a 12-factor app principles (Bogner et al., 2019) es reportada explícitamente como otra práctica en la literatura.

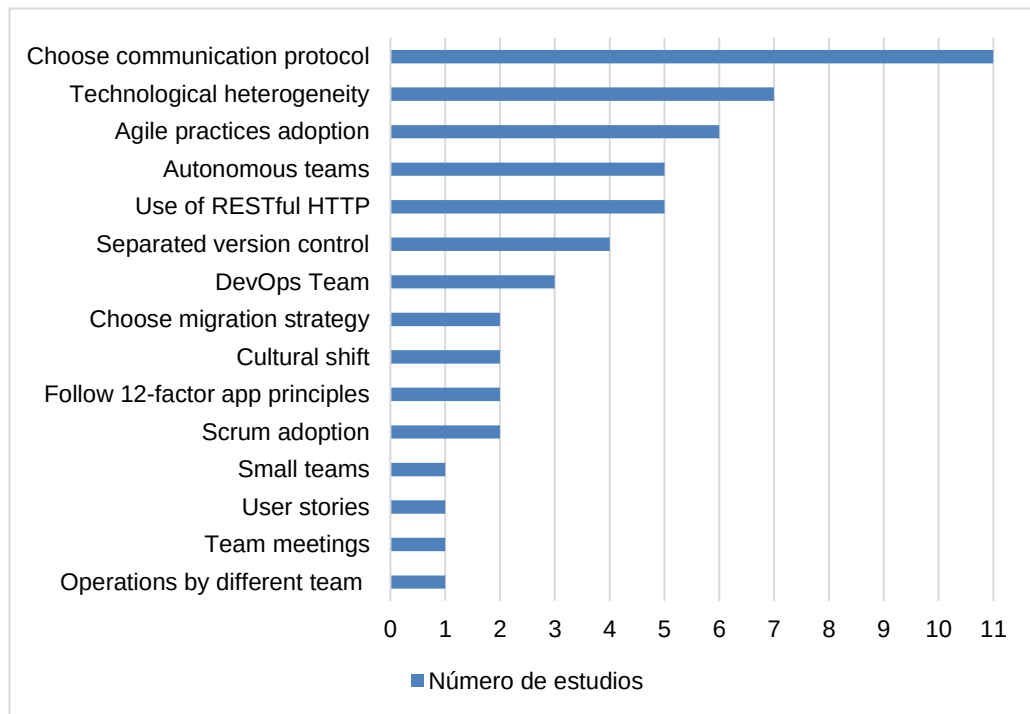


Figura 7 Prácticas de los equipos reportadas

Una fase del desarrollo en donde se encuentran prácticas que involucran directamente a los equipos, es la de diseño. Al diseñar un sistema con arquitectura de microservicios los desarrolladores se encontrarán con una tarea que se repetirá sin importar las características del dominio, la descomposición. En este sentido, se identificaron técnicas que los desarrolladores ocupan para lograr desacoplar un sistema en partes pequeñas, en la Figura 8 se muestran las técnicas detectadas en la literatura. En primer lugar, los estudios reportan las siguientes estrategias para lograr una descomposición adecuada: database decomposition (Bogner et al., 2019; Fan & Ma, 2017; Garriga, 2018; Lotz et al., 2019; Salah et al., 2016; Srikaew & Kim, 2017), system decomposition (Buchgeher et al., 2017; Di Francesco et al., 2018) y domain decomposition (Di Francesco et al., 2018; Sousa et al., 2020). En casos donde el sistema era muy complejo para lograr descomponerlo se optó por una reescritura completa de la aplicación (Bogner et al., 2019; Fritzsch et al., 2019). De igual forma la técnica de descomposición funcional (Fritzsch et al., 2019; Idoughi et al., 2017; Srikaew & Kim, 2017) es mencionada para lograr descomponer un sistema con base en las funcionalidades principales, ya que es una estrategia que analiza de manera minuciosa un proceso complejo para examinar sus elementos individuales. Finalmente, horizontal decomposition (Zhang et al., 2019), vertical decomposition (Zhang et al., 2019), featured-oriented decomposition (Bogner et al., 2019), static analysis (SA) (Ponce et al., 2019) y dynamic analysis (Ponce et al., 2019) son técnicas que también se mencionan en la literatura.

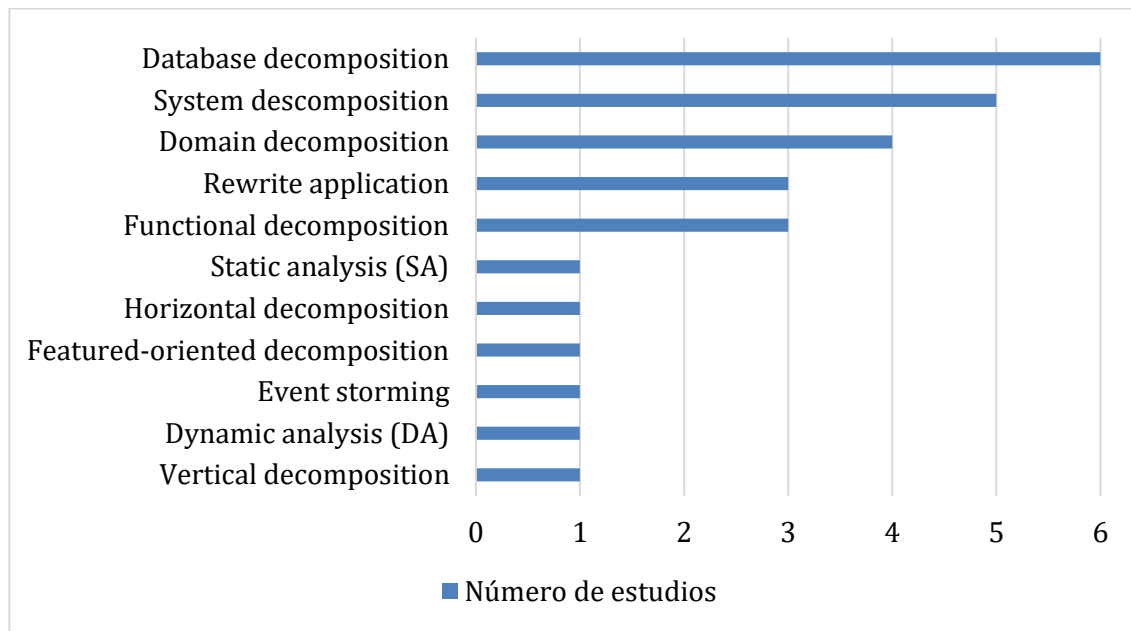


Figura 8 Técnicas de descomposición

En la Figura 9 se muestran diversos enfoques de diseño que las organizaciones adoptan para lograr el desarrollo de microservicios. Dichos enfoques están conformados por las diversas estrategias y patrones de diseño que son utilizados. Domain Driven Design al ser un enfoque que busca dividir un dominio complejo en subdominios, es mencionado en la literatura como una de las alternativas principales durante el diseño de microservicios (Bogner et al., 2019; Di Francesco et al., 2018; Fan & Ma, 2017; Fritzsche et al., 2019; Ghofrani & Bozorgmehr, 2019; Waseem et al., 2020; Zhang et al., 2019). De igual forma el uso de patrones de diseño es reportado con frecuencia. Model-Driven Design (MDD) (Ghofrani & Bozorgmehr, 2019; Ponce et al., 2019) también es mencionado debido a que usa elementos de diseño como entrada, dichos elementos pueden ser: capacidades de negocio, objetos de negocio, entidades de dominio y diagrama de flujo de datos de la lógica de negocio. También, para desacoplar la base de datos se utilizan enfoques como una base de datos por servicio (Kalske et al., 2018; Soldani et al., 2018; Waseem et al., 2020) y administración descentralizada de los datos. Por último, design for failure (Buchgeher et al., 2017; Zhang et al., 2019) es reportado en algunas ocasiones en la literatura, concepto introducido originalmente por Martin Fowler sobre diseñar microservicios pensando en que fallarán y que acciones tomar.

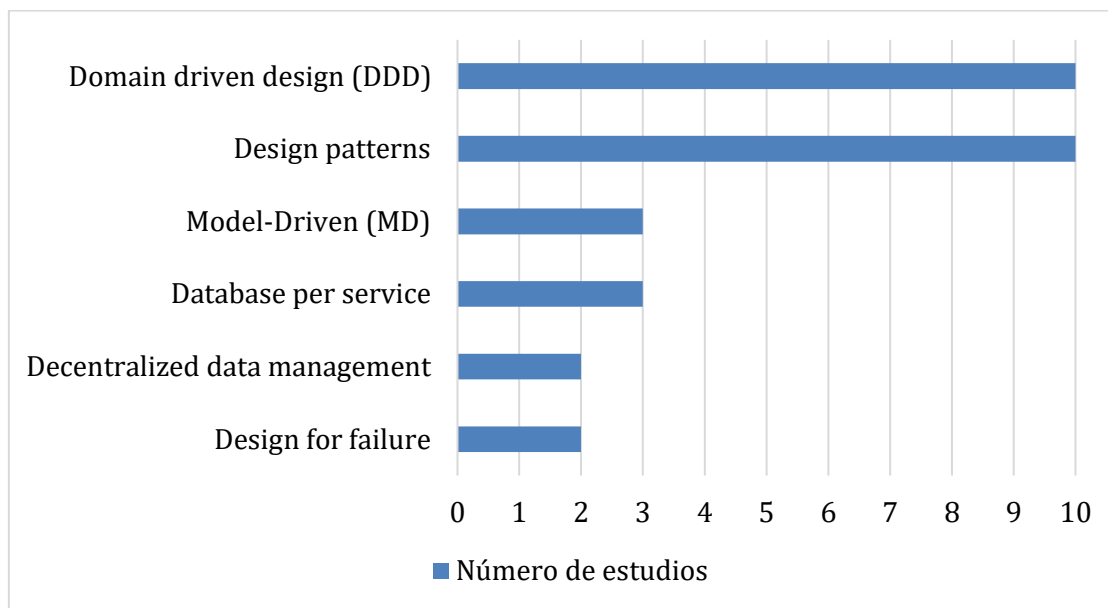


Figura 9 Enfoques de diseño

De igual forma, en la Figura 10 se pueden observar prácticas relacionadas con el despliegue y la operación de los microservicios. El uso de contenedores es muy recurrente durante este tipo de desarrollo, debido a la integración natural que tienen los microservicios con las plataformas basadas en contenedores, cada microservicio puede ser empaquetado y entregado en su propio contenedor (Bogner et al., 2019; Di Francesco et al., 2019; Fan & Ma, 2017; Lotz et al., 2019; Luz et al., 2018; O'Connor et al., 2016; Salah et al., 2016; Soldani et al., 2018; Sousa et al., 2020; Srikaew & Kim, 2017; Waseem et al., 2020). También, se reportan prácticas que tienen una estrecha relación con una cultura DevOps; Continuous integration (CI) (Bogner et al., 2019; Fan & Ma, 2017; Fritzsche et al., 2019; Hassan et al., 2020; Idoughi et al., 2017; Premchand & Choudhry, 2018; Sousa et al., 2020; Zhang et al., 2019), Continuous delivery (CD), Continuous deployment (Bogner et al., 2019; Fan & Ma, 2017; Fritzsche et al., 2019; O'Connor et al., 2016; Premchand & Choudhry, 2018; Zhang et al., 2019), al ejecutar estas actividades las organizaciones pueden lograr la integración de diversos microservicios, software listo para ser entregado al usuario y el despliegue automatizado del sistema. Después, el monitoring y logging es mencionado en diversos estudios (Ahmad et al., 2018; Di Francesco et al., 2018; Fan & Ma, 2017; Garriga, 2018; Hassan et al., 2020; Zhang et al., 2019), debido a que los microservicios se comunican por la red mediante sus APIs, deben de estar en constante monitoreo para asegurar su funcionamiento correcto. Finalmente, existen organizaciones que se apegan al paradigma de computación en la nube (Ahmad et al., 2018; Bogner et al., 2019; Di Francesco et al., 2018; Garriga, 2018; Soldani et al., 2018), entregando sus microservicios en contenedores a un proveedor de servicios y existen otras que reportan un grado de automatización únicamente en la etapa de despliegue (Buchgeher et al., 2017; Fan & Ma, 2017; Luz et al., 2018; Premchand & Choudhry, 2018).

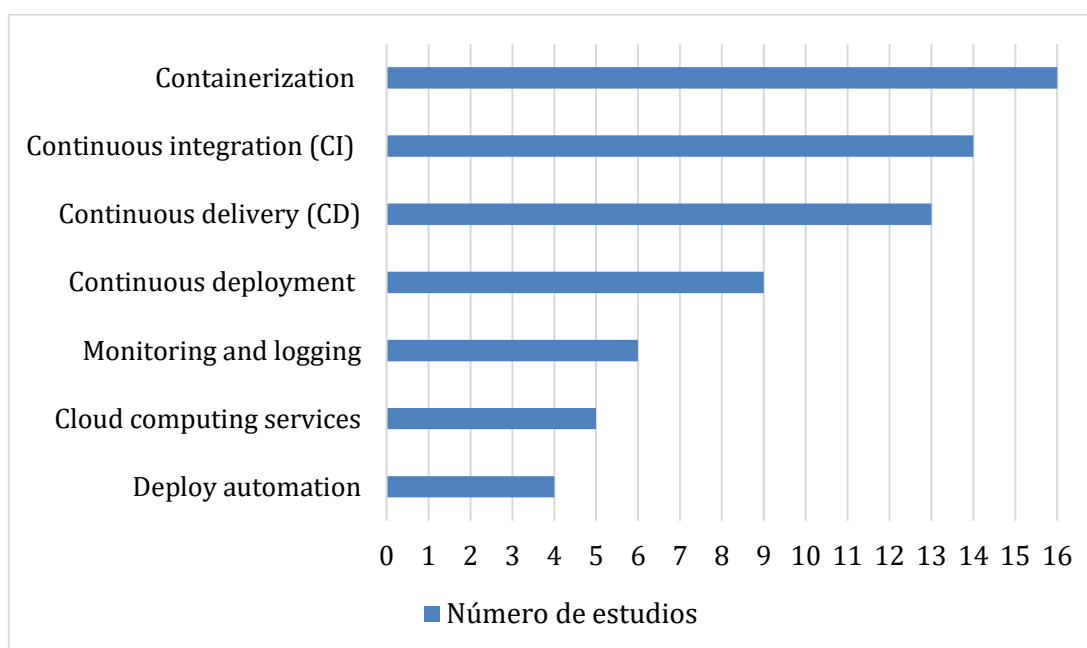


Figura 10 Despliegue y operaciones

6.1.3 Organización de los equipos de desarrollo

La organización de los equipos es de suma importancia en cualquier tipo de desarrollo, pero específicamente en el de microservicios la arquitectura del sistema está estrechamente relacionada con dicha organización. Los modelos de organización hallados son las pautas que los equipos respetan para la división de responsabilidades durante el desarrollo. Debido a la naturaleza del estilo arquitectónico se reporta que un modelo de gobernanza descentralizada es utilizado para una adopción satisfactoria (Bogner et al., 2019; Buchgeher et al., 2017; Fritzsche et al., 2019; Hassan et al., 2020; Premchand & Choudhry, 2018; Zhang et al., 2019). A diferencia de un modelo centralizado que permite una plataforma tecnológica única, los microservicios admiten una mezcla de múltiples lenguajes, marcos de desarrollo y tecnologías de datos, es decir, la diversidad tecnológica (Fan & Ma, 2017; Lotz et al., 2019; Taibi, Lenarduzzi, Pahl, et al., 2017; Zhang et al., 2019). Dicho modelo descentralizado va de la mano con los equipos autónomos que pueden tomar sus propias decisiones y la heterogeneidad tecnológica (Bogner et al., 2019). En segundo lugar, se reportan modelos alrededor de conceptos del negocio y que se comuniquen a través de APIs (Ahmad et al., 2018; Di Francesco et al., 2018; Garriga, 2018; Gundelsby, 2018; Premchand & Choudhry, 2018), esto permite que los equipos tengan la responsabilidad de un conjunto de componentes y puedan comunicarse con otro mediante una API. Siguiendo esta línea se puede observar que los modelos que fomentan la independencia de los procesos son los más reportados: despliegue independiente (Soldani et al., 2018), ciclo de vida independiente (Buchgeher et al., 2017; Idoughi et al., 2017; Kalske et al., 2018; Lotz et al., 2019; Soldani et al., 2018; Srikaew & Kim, 2017; Zhang et al., 2019). La posibilidad de un despliegue independiente es reconocido como uno de los mayores beneficios en el desarrollo con arquitectura de microservicios (Soldani et al., 2018). Existen algunos dominios con características en particular, ahí fue de suma importancia adoptar un modelo de gobernanza centralizada (Bogner et al., 2019), a pesar de que existe un grado considerable de autonomía, las decisiones finales eran responsabilidad de un equipo primario. De igual forma también se reporta que algunas organizaciones optan por un greenfield approach que indica la libertad de la reconstrucción total de un sistema (Bogner et al., 2019). Después se tiene conway's law mencionado explícitamente, dicha ley impone muchos principios mencionados anteriormente para lograr una independencia satisfactoria (Kalske et al., 2018; Zhang et al., 2019). En la Figura 11 se muestran los diversos modelos de organización y el número de estudios.

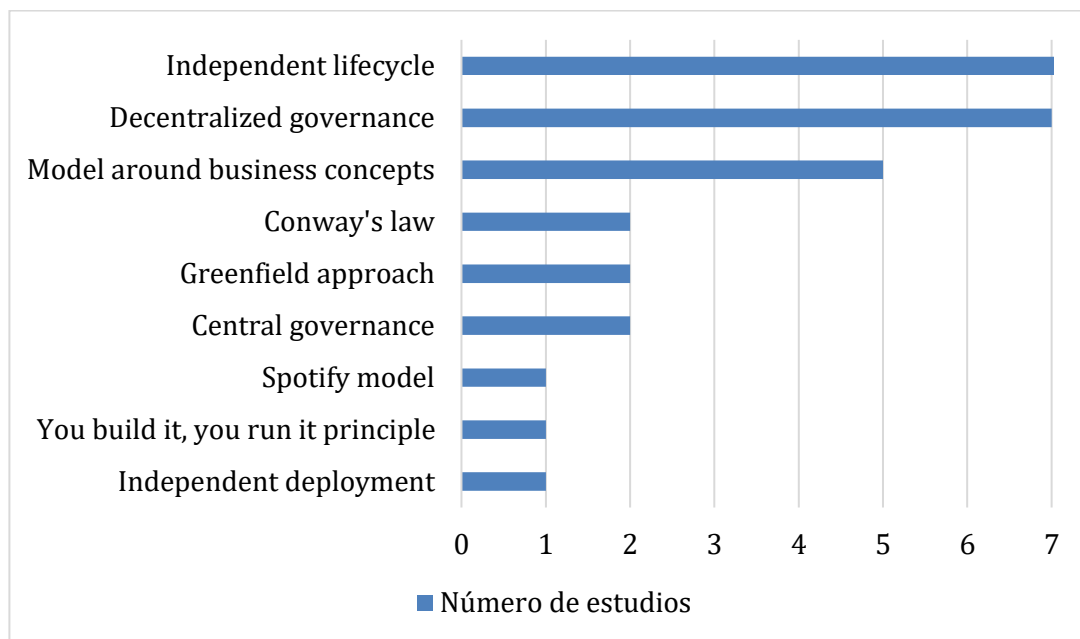


Figura 11 Modelos de organización

6.1.4 Retos que conllevan las prácticas de los equipos en el desarrollo

Tomar la decisión de adoptar una arquitectura de microservicios conlleva varios factores que tienen que enfrentar los equipos de desarrollo. El cambio que existe no solamente impacta nivel técnico o de diseño, también implica una transformación organizacional como se ha mencionado anteriormente. Dicho cambio en las organizaciones trae consigo retos en las habilidades de los desarrolladores que se pueden observar en la Figura 12. De primera instancia cuando las organizaciones quieren migrar a microservicios, se encuentran con la necesidad de tener equipos integrados por desarrolladores con habilidades específicas o práctica con este tipo de sistemas (Bogner et al., 2019; Buchgeher et al., 2017; Fritzsche et al., 2019; Ghofrani & Bozorgmehr, 2019; Kalske et al., 2018; Luz et al., 2018; O'Connor et al., 2016; Ponce et al., 2019; Sousa et al., 2020; Taibi, Lenarduzzi, & Pahl, 2017; Waseem & Liang, 2017; Zhang et al., 2019). El cambio de mentalidad también es uno de los retos que se enfrentan los desarrolladores (Ahmad et al., 2018; Bogner et al., 2019; Di Francesco et al., 2018; Fritzsche et al., 2019; Ghofrani & Bozorgmehr, 2019; Kalske et al., 2018; Ponce et al., 2019), debido a que el estilo arquitectónico con el que suelen trabajar implica un flujo de trabajo diferente al de microservicios. También se enfrentan retos que tienen que ver con la complejidad de los sistemas (Di Francesco et al., 2019, 2018; Fan & Ma, 2017; Garriga, 2018; Lotz et al., 2019), siguiendo esta línea, igual se encuentran con la dificultad a la hora de las pruebas (Ghofrani & Bozorgmehr, 2019; Soldani et al., 2018; Taibi, Lenarduzzi, & Pahl, 2017; Waseem & Liang, 2017). Otro de los retos hallados es la necesidad de desarrolladores con experiencia en el desarrollo basado en servicios (Buchgeher et al., 2017; Zhang et al., 2019), Zhang et al. Reporta que la decisión de que partes de un monolito deberían separarse en servicios, fueron tomadas por desarrolladores con experiencia en el área (Buchgeher et al., 2017). Al final se encuentran retos que tienen que ver con la coordinación de los equipos de desarrollo (Gundelsby, 2018; Salah et al., 2016) y la estructura de la organización. Los equipos de desarrollo deben sincronizarse de manera adecuada, para no interferirse durante el trabajo y que cause una baja productividad (Premchand & Choudhry, 2018).

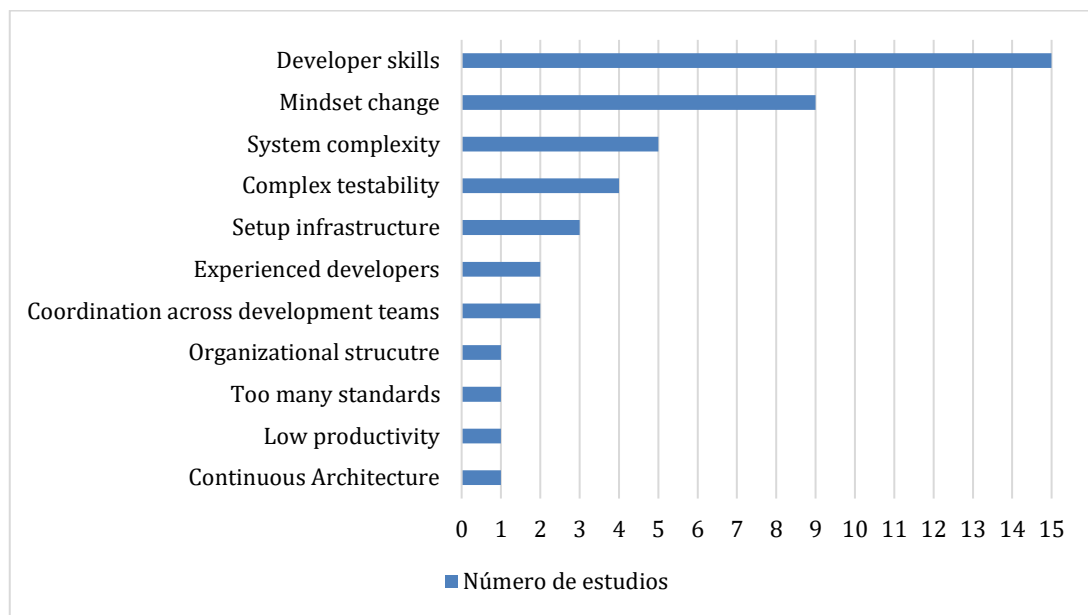


Figura 12 Retos relacionados con las habilidades

Como se mencionó anteriormente, la descomposición durante el desarrollo de microservicios es una de las prácticas más frecuentes, pero lograr una descomposición exitosa conlleva sus retos. En 12 estudios se reporta como un problema encontrar la estrategia de descomposición adecuada, Luz et al. Reportaron que el principal reto enfrentado durante la adopción es la capacidad de entender cual es la manera correcta de descomponer un monolito (Luz et al., 2018). Por otra parte, al tener varios microservicios se encuentra el reto de saber cual es el límite de cada uno (Di Francesco et al., 2018; Garriga, 2018; Gundelsby, 2018; Luz et al., 2018; Ponce et al., 2019; Soldani et al., 2018) este punto tiene que ver con la granularidad de los servicios, identificar que tan grande o que tan pequeño debe de ser, no es una tarea fácil. En organizaciones que tienen sistemas grandes o muy complejos, la capacidad de entenderlos es otro de los problemas que se reportan en la literatura (Di Francesco et al., 2018; Fritsch et al., 2019; Zhang et al., 2019). En este sentido, también se puede hallar el alto acoplamiento que pueden tener los componentes existentes de un sistema (Di Francesco et al., 2018)(Salah et al., 2016). Por último, una descomposición inadecuada puede traer consigo una independencia caótica, esto significa una serie de retos durante todo el ciclo de vida, desde la incapacidad para poder agregar funciones sin afectar a otros servicios, hasta el despliegue de los mismos, en este punto también se encuentra lo que es la dificultad de integrar los diversos microservicios. La Figura 13 muestra los retos mencionados anteriormente.

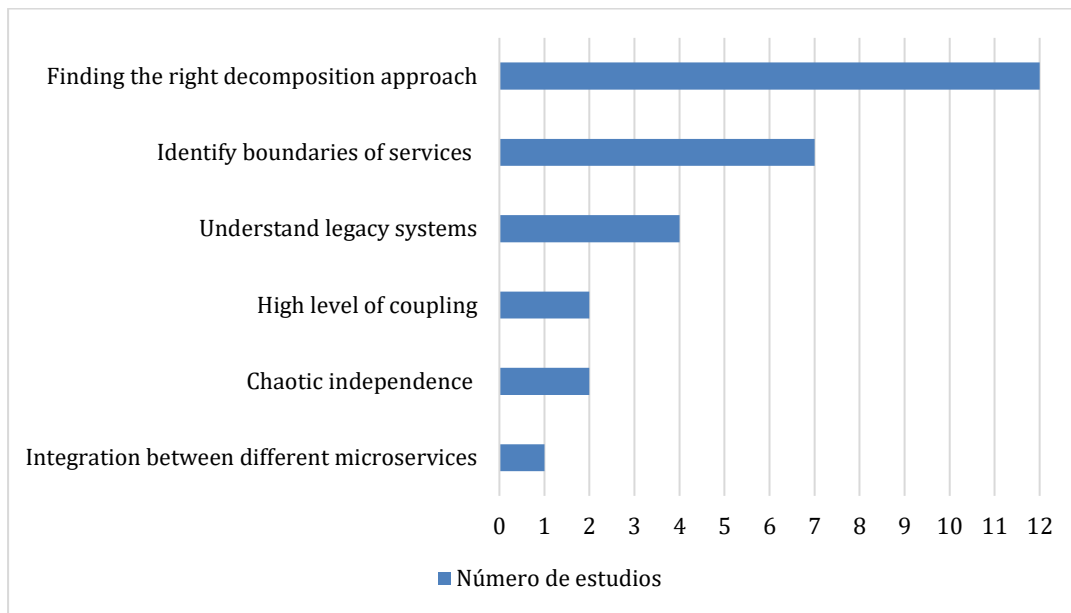


Figura 13 Retos en la descomposición

7. DISCUSIÓN

Los resultados de la revisión de la literatura muestran que la arquitectura de microservicios es un estilo arquitectónico que han adoptado las organizaciones durante los últimos años para resolver las problemáticas en sus dominios específicos. Para la inclusión de estudios, los años 2015 al 2020 fueron definidos ya que es cuando el tema de microservicios empezó a tener auge. A pesar de que el año 2020 cuenta con un número bajo de estudios recuperados, esto sucede debido a los criterios que se tienen que cumplir para ser incluidos en esta investigación, sin embargo, en la literatura se encuentran diversos trabajos que abordan el tema de microservicios y apunta a que seguirán en aumento. Se analizaron los estudios para responder las preguntas de investigación, los datos relevantes fueron extraídos con una herramienta para análisis de datos cualitativos. Dicha herramienta también fue utilizada durante la síntesis de información y posteriormente exportados a una hoja de cálculo para facilitar su lectura. Con base en la información recolectada de los estudios, se analizó el estado del arte de los equipos de desarrollo con arquitectura de microservicios. Para responder la PI-1 se recolectaron características de los equipos mencionadas explícitamente en los estudios, los resultados muestran que las particularidades que ellos poseen están relacionadas con el problema a resolver y el modelo de organización que adopten. Los estudios abordan de manera implícita detalles de los equipos, mencionando que deben de ser pequeños y tener la suficiente capacidad cognitiva para desarrollar y mantener el servicio a su cargo. Para la PI-2 entre las prácticas reportadas se encuentran actividades que los equipos tienen que realizar con el cambio de paradigma, ya que la adopción de microservicios trae consigo un cambio cultural. De igual manera la descomposición es un tema recurrente, debido a que para adoptar este estilo arquitectónico debe de existir una separación de responsabilidades. Tomando en cuenta lo anterior los equipos optan por diversos enfoques de diseño que sirven para lograr un desarrollo exitoso. En cuanto a modelos de organización como respuesta a la PI-3 se encuentra dominante la descentralización y la independencia durante el ciclo de vida del servicio. El tener un sistema separado por capacidades de negocio se ajusta a los modelos independientes. Finalmente, entre los retos que responden la PI-4 se encuentra el factor humano, debido a la dificultad de adaptación, falta de habilidades y nuevas actividades que implican un cambio de mentalidad en los desarrolladores. De igual manera, durante la descomposición se encuentran diversos retos que deben de enfrentar los equipos de desarrollo, debido a que los sistemas pueden tener un alto nivel de acoplamiento o una alta complejidad que conlleva un serie de barreras.

Los hallazgos mencionados son relevantes para las personas que deseen investigar a profundidad acerca de la organización de los equipos en los microservicios, que prácticas realizan y que retos se pueden encontrar durante su desarrollo, ya que esta es una fuente bibliográfica en donde se recopila el estado del arte de dicho tema. Por último, cabe destacar que gran parte de los resultados obtenidos, no habían sido mencionados por trabajos relacionados y eso es debido a que el objetivo de este estudio fue analizar las diversas prácticas de los equipos de desarrollo de microservicios y hasta este momento no existía un estudio que abordara el tema de manera rigurosa como se realizó en este trabajo.

8. CONCLUSIÓN

En este estudio se buscó analizar las prácticas de los equipos que desarrollan sistemas con una Arquitectura de Microservicios. Las preguntas de investigación que se definieron se centran en las prácticas de los equipos que desarrollan este tipo de sistemas, así como los retos a los que se enfrentan. Se seleccionaron 26 estudios primarios que sirvieron para realizar una síntesis temática e identificar diferentes patrones. Entre uno de los descubrimientos se encuentra la tendencia de estudios relacionados con el desarrollo de microservicios. Las características de los equipos de desarrollo concuerdan con premisas ya mencionadas, pero dependiendo el dominio específico tienen que adaptarse. Se encontraron diversas prácticas que los equipos de desarrollo deben realizar en el ámbito cultural, en su forma de trabajo y durante la descomposición del sistema. De igual forma se resalta la importancia de un modelo de organización que brinde un nivel de independencia deseado. También es importante considerar todos los retos que conlleva el desarrollo con microservicios, debido a que el factor humano juega un papel muy importante respecto a las habilidades y cambio de mentalidad necesario. Los hallazgos que se han detectado pueden servir para diversas líneas de investigación como: El análisis de las estrategias de descomposición y cual es la adecuada, profundizar en las prácticas DevOps que proporcionan el mayor beneficio a los microservicios, proponer un marco de trabajo para estandarizar la organización del desarrollo de manera explícita. Los resultados que aquí se presentan pueden crear conciencia en profesionales del desarrollo de software sobre los cambios en las prácticas y organización de los equipos que el desarrollo de Microservicios trae consigo, así como los retos a los que se tendrían que enfrentar en la adopción de este estilo arquitectónico. Para los investigadores, nuestro estudio puede ser una base para estudios que deseen profundizar en los aspectos a nivel organizacional en la adopción del desarrollo de sistemas con una Arquitectura de Microservicios y proponer soluciones a los diferentes retos identificados. Por lo anterior, se ve como trabajo futuro un análisis de literatura gris, así como otro tipo de estudios empíricos, incluyendo encuestas y entrevistas, que profundicen en los hallazgos aquí descritos, así como el desarrollo de guías y otro tipo de artefactos que aborden los retos aquí descritos.

9. REFERENCIAS

- Ahmad, N., Naveed, Q. N., & Hoda, N. (2018). Strategy and procedures for Migration to the Cloud Computing. 2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS), 1–5. <https://doi.org/10.1109/ICETAS.2018.8629101>
- AWS. (2020). Deployment Strategies - Introduction to DevOps on AWS.
- Bogner, J., Fritzsche, J., Wagner, S., & Zimmermann, A. (2019). Microservices in Industry: Insights into Technologies, Characteristics, and Software Quality. 2019 IEEE International Conference on Software Architecture Companion (ICSA-C), 187–195. <https://doi.org/10.1109/ICSA-C.2019.00041>
- Bucchiarone, A., Dragoni, N., Dustdar, S., Larsen, S. T., & Mazzara, M. (2018). From Monolithic to Microservices: An Experience Report from the Banking Domain. IEEE Software, 35(3), 50–55. <https://doi.org/10.1109/MS.2018.2141026>
- Buchgeher, G., Winterer, M., Weinreich, R., Luger, J., Wingelhofer, R., & Aistleitner, M. (2017). Microservices in a Small Development Organization. En A. Lopes & R. de Lemos (Eds.), Software Architecture (pp. 208–215). Springer International Publishing. https://doi.org/10.1007/978-3-319-65831-5_15
- Conway, M. E. (1968). How do committees invent. Datamation, 14(4), 28–31.
- Cruzes, D. S., & Dyba, T. (2011). Recommended Steps for Thematic Synthesis in Software Engineering. 2011 International Symposium on Empirical Software Engineering and Measurement, 7491, 275–284. <https://doi.org/10.1109/ESEM.2011.36>
- Di Francesco, P., Lago, P., & Malavolta, I. (2019). Architecting with microservices: A systematic mapping study. Journal of Systems and Software, 150(April), 77–97. <https://doi.org/10.1016/j.jss.2019.01.001>
- Di Francesco, P., Lago, P., & Malavolta, I. (2018). Migrating Towards Microservice Architectures: An Industrial Survey. 2018 IEEE International Conference on Software Architecture (ICSA), 29–2909. <https://doi.org/10.1109/ICSA.2018.00012>
- Dieste, O., Grimán, A., & Juristo, N. (2009). Developing search strategies for detecting relevant experiments. Empirical Software Engineering, 14(5), 513–539. <https://doi.org/10.1007/s10664-008-9091-7>
- Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. Information and Software Technology, 50(9), 833–859. <https://doi.org/https://doi.org/10.1016/j.infsof.2008.01.006>
- Fan, C.-Y., & Ma, S.-P. (2017). Migrating Monolithic Mobile Application to Microservice Architecture: An Experiment Report. 2017 IEEE International Conference on AI & Mobile Services (AIMS), 109–112. <https://doi.org/10.1109/AIMS.2017.23>
- Fowler, S. J. (2016). Production-Ready Microservices: Building Standardized Systems Across an Engineering Organization (1st ed.). O'Reilly Media, Inc.
- Francesco, P. Di, Malavolta, I., & Lago, P. (2017). Research on Architecting Microservices: Trends, Focus, and Potential for Industrial Adoption. 2017 IEEE International Conference on Software Architecture (ICSA), 21–30. <https://doi.org/10.1109/ICSA.2017.24>

- Fritzsche, J., Bogner, J., Wagner, S., & Zimmermann, A. (2019). Microservices Migration in Industry: Intentions, Strategies, and Challenges. 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME), 481–490. <https://doi.org/10.1109/ICSME.2019.00081>
- Garriga, M. (2018). Towards a Taxonomy of Microservices Architectures. En A. Cerone & M. Roveri (Eds.), *Software Engineering and Formal Methods* (pp. 203–218). Springer International Publishing. https://doi.org/10.1007/978-3-319-74781-1_15
- Ghofrani, J., & Bozorgmehr, A. (2019). Migration to Microservices: Barriers and Solutions. En H. Florez, M. Leon, J. M. Diaz-Nafria, & S. Belli (Eds.), *Applied Informatics* (pp. 269–281). Springer International Publishing. https://doi.org/10.1007/978-3-030-32475-9_20
- Gundelsby, J. H. (2018). Enabling autonomous teams in large-scale agile through architectural principles. *Proceedings of the 19th International Conference on Agile Software Development: Companion*, 1–4. <https://doi.org/10.1145/3234152.3234183>
- Hassan, S., Bahsoon, R., & Kazman, R. (2020). Microservice transition and its granularity problem: A systematic mapping study. *Software: Practice and Experience*, 50(9), 1651–1681. <https://doi.org/10.1002/spe.2869>
- Idoughi, D., Abdelouhab, K. A., & Kolski, C. (2017). Towards a microservices development approach for the crisis management field in developing countries. 2017 4th International Conference on Information and Communication Technologies for Disaster Management (ICT-DM), 1–6. <https://doi.org/10.1109/ICT-DM.2017.8275679>
- Kalske, M., Mäkitalo, N., & Mikkonen, T. (2018). Challenges When Moving from Monolith to Microservice Architecture. En I. Garrigós & M. Wimmer (Eds.), *Current Trends in Web Engineering* (pp. 32–47). Springer International Publishing. https://doi.org/10.1007/978-3-319-74433-9_3
- Larrucea, X., Santamaria, I., Colomo-Palacios, R., & Ebert, C. (2018). Microservices. *IEEE Software*, 35(3), 96–100. <https://doi.org/10.1109/MS.2018.2141030>
- Lotz, J., Vogelsang, A., Benderius, O., & Berger, C. (2019). Microservice Architectures for Advanced Driver Assistance Systems: A Case-Study. 2019 IEEE International Conference on Software Architecture Companion (ICSA-C), 45–52. <https://doi.org/10.1109/ICSA-C.2019.00016>
- Luz, W., Agilar, E., de Oliveira, M. C., de Melo, C. E. R., Pinto, G., & Bonifácio, R. (2018). An experience report on the adoption of microservices in three Brazilian government institutions. *Proceedings of the XXXII Brazilian Symposium on Software Engineering - SBES '18*, 32–41. <https://doi.org/10.1145/3266237.3266262>
- O'Connor, R. V., Elger, P., & Clarke, P. M. (2016). Exploring the impact of situational context. *Proceedings of the International Conference on Software and Systems Process*, 6–10. <https://doi.org/10.1145/2904354.2904368>
- Pahl, C., & Jamshidi, P. (2016). Microservices: A Systematic Mapping Study. *Proceedings of the 6th International Conference on Cloud Computing and Services Science*, 1(Closer), 137–146. <https://doi.org/10.5220/0005785501370146>
- Ponce, F., Márquez, G., & Astudillo, H. (2019). Migrating from monolithic architecture to microservices: A Rapid Review. 2019 38th International Conference of the Chilean Computer Science Society (SCCC), 1–7. <https://doi.org/10.1109/SCCC49216.2019.8966423>

Premchand, A., & Choudhry, A. (2018). Architecture Simplification at Large Institutions using Micro Services. 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT), 30–35. <https://doi.org/10.1109/IC3IoT.2018.8668173>

Salah, T., Jamal Zemerly, M., Chan Yeob Yeun, Al-Qutayri, M., & Al-Hammadi, Y. (2016). The evolution of distributed systems towards microservices architecture. 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST), 318–325. <https://doi.org/10.1109/ICITST.2016.7856721>

Soldani, J., Tamburri, D. A., & Van Den Heuvel, W.-J. (2018). The pains and gains of microservices: A Systematic grey literature review. Journal of Systems and Software, 146, 215–232. <https://doi.org/10.1016/j.jss.2018.09.082>

Sousa, P. S. de, Nogueira, N. P., Santos, R. C. dos, Maia, P. H. M., & Souza, J. T. de. (2020). Building a prototype based on Microservices and Blockchain technologies for notary's office: An academic experience report. 2020 IEEE International Conference on Software Architecture Companion (ICSA-C), 122–129. <https://doi.org/10.1109/ICSA-C50368.2020.00031>

Srikaew, P., & Kim, I. (2017). A microservice development for document management system. 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), 1–4. <https://doi.org/10.1109/CAIPT.2017.8320721>

Taibi, D., Lenarduzzi, V., & Pahl, C. (2017). Processes, Motivations, and Issues for Migrating to Microservices Architectures: An Empirical Investigation. IEEE Cloud Computing, 4(5), 22–32. <https://doi.org/10.1109/MCC.2017.4250931>

Taibi, D., Lenarduzzi, V., Pahl, C., & Janes, A. (2017). Microservices in agile software development. 1–5. <https://doi.org/10.1145/3120459.3120483>

Wang, Y., & Kadiyala, H. (2020). Promises and Challenges of Microservices : an Exploratory Study. 1–45.

Waseem, M., & Liang, P. (2017). Microservices Architecture in DevOps. 2017 24th Asia-Pacific Software Engineering Conference Workshops (APSECW), 13–14. <https://doi.org/10.1109/APSECW.2017.18>

Waseem, M., Liang, P., & Shahin, M. (2020). A Systematic Mapping Study on Microservices Architecture in DevOps. Journal of Systems and Software, 170, 110798. <https://doi.org/10.1016/j.jss.2020.110798>

Zhang, H., Li, S., Jia, Z., Zhong, C., & Zhang, C. (2019). Microservice Architecture in Reality: An Industrial Inquiry. 2019 IEEE International Conference on Software Architecture (ICSA), 51–60. <https://doi.org/10.1109/ICSA.2019.00014>

NOTAS BIOGRÁFICAS



Alan González Heredia actualmente es estudiante de Ingeniería de Software en la Universidad Veracruzana, se encuentra realizando prácticas profesionales e impulsando prácticas de Ingeniería de Software en Intel Corporation. Sus áreas de interés son la Ingeniería de Software, el Diseño y la Arquitectura de software.



Jorge Octavio Ocharán Hernández es Doctor en Ciencias de la Computación por la Universidad Veracruzana y Maestro en Ingeniería de Software por la misma institución, posee además el título de Máster en Software Libre: Administrador de Redes y Sistemas Operativos por la Universitat Oberta de Catalunya y de Ingeniero en Tecnologías Estratégicas de la Información por la Universidad Anáhuac Campus Veracruz. Actualmente es profesor de tiempo completo de la Facultad de Estadística e Informática de la Universidad Veracruzana en donde realiza actividades de docencia, tutoría, vinculación e investigación desde 2017. Es miembro de la Association for Computing Machinery y de la IEEE Computer Society. Sus áreas de interés son la Ingeniería de Software, el Diseño y la Arquitectura de Software, la Ingeniería de Requisitos y los Aspectos Humanos de la Ingeniería de Software. Es un entusiasta del Software Libre y de Fuente Abierta.



María de los Ángeles Arenas Valdés es Maestra en Ciencias Computacionales por la Fundación Arturo Rosenblueth, México. Licenciada en Informática por la Universidad Veracruzana. Actualmente es profesor en la Facultad de Estadística e Informática de la Universidad Veracruzana. Sus intereses se centran en temas relacionados con la Ingeniería de Software, en especial con la Ingeniería de Requisitos.



Karen Cortés Verdín es Doctora en Ciencias de la Computación Doctorado y Maestra en Ingeniería de Software por el Centro de Investigación en Matemáticas (CIMAT) A.C. en Guanajuato México. M.Sc Information Systems Engineering, por la University of Manchester Institute of Science and Technology (UMIST), Manchester Reino Unido. Actualmente es profesora de tiempo completo en la Licenciatura en Ingeniería de Software de la Universidad Veracruzana (UV), en Xalapa Veracruz México. Cuenta con diplomados en Enseñanza Superior, Propiedad Intelectual, Redes, Diseño Curricular y Habilidades del Pensamiento. Sus áreas de interés son Líneas de Productos de Software, Arquitecturas de Software, Orientación a Aspectos, Orientación a Concerns, Calidad de Software y Procesos de Software.



Esta obra está bajo una licencia de Creative Commons
Atribución-NoComercial-CompartirIgual 4.0 Internacional.