



NUEVAS REALIDADES PARA LA EDUCACIÓN EN INGENIERÍA:
CURRÍCULO, TECNOLOGÍA, MEDIO AMBIENTE Y DESARROLLO

13 - 16
DE SEPTIEMBRE

2022

CARTAGENA DE INDIAS,
COLOMBIA



Encuentro Internacional de
Educación en Ingeniería ACOFI

Modelos transaccionales avanzados como alternativa para la implementación de transacciones de larga duración en microservicios

Christian Andrés Candela Uribe

**Universidad del Quindío
Armenia, Colombia**

Resumen

La arquitectura de microservicios (MSA) ha ganado reconocimiento y aceptación en la industria del software, usándose cada vez más en el desarrollo de nuevos sistemas (Alshuqayran et al., 2016; Eismann et al., 2020; Marquez et al., 2018). Algunos de los motivos de su popularidad son: una base de código más pequeña que facilita el desarrollo, prueba e implementación de código más rápidos, así como la optimización de la plataforma en función del tipo de microservicio, soporte para equipos de desarrollo independientes y la capacidad de escalar cada componente independientemente (Chandramouli, 2019; Hassan et al., 2019).

Sin embargo, la MSA trae consigo múltiples desafíos, entre ellos, las operaciones transaccionales donde participan múltiples microservicios representan un reto latente a las organizaciones que desean adoptar una arquitectura basada en microservicios (Di Francesco et al., 2019; Knoche & Hasselbring, 2019; Waseem et al., 2020). Lo anterior, debido a que en MSA dichas transacciones pueden involucrar bases de datos (algunas incluso sin soporte transaccional) y elementos de comunicación asíncrona. Estos últimos con duración desconocida (transacciones de larga duración) lo que hace inviable el bloqueo de registros dificultando así conocer la verdad del estado del sistema (Elmagarmid, 1992; Knoche & Hasselbring, 2019).

Considerando lo expuesto anteriormente, no todas las transacciones de negocio pueden cumplir con las propiedades ACID (Dalal et al., 2003; Gray, 1981; Papazoglou, 2003; Papazoglou & Kratz, 2007; Sun & Aiello, 2007; Tang et al., 2011; Tang et al., 2006), por lo que podemos concluir que en operaciones transaccionales que abarcan más de un microservicios el modelo transaccional tradicional no es viable, siendo necesario plantear alternativas que permitan la realización de estas operaciones. Es posible buscar una alternativa en el conjunto de Modelos

Transaccionales Avanzados (ATM – Advance Transactional Model) o Modelos Transaccionales Extendidos, los cuales relajan las propiedades ACID, no cumpliendo una o algunas de dichas propiedades, para soportar transacciones de larga duración (Elmagarmid, 1992; Torres et al., 2009).

Este trabajo busca adaptar un MTA para su uso en MSA como alternativa para la implementación de operaciones transaccionales que involucran múltiples microservicios. Para lograrlo, metodológicamente se plantea 5 etapas: Planteamiento del objetivo, Reunión y organización de información, Diseño, Evaluación y Presentación de resultados. Estas etapas se basan en el método general de investigación referido por Sánchez (2004), además de elementos del flujo de trabajo de la investigación cualitativa descritos por Hernández-Sampieri and Torres (2018).

Palabras clave: microservicios; MSA; modelos transaccionales avanzados; transacciones de larga duración

Abstract

Microservice architecture (MSA) has gained recognition and acceptance in the software industry, being increasingly used in the development of new systems (Alshuqayran et al., 2016; Eismann et al., 2020; Marquez et al., 2018). Some of the reasons for its popularity include a smaller code base that facilitates faster code development, testing, and deployment, as well as platform optimization based on the type of microservice, support for independent development teams, and the ability to scale each component independently (Chandramouli, 2019; Hassan et al., 2019).

However, MSA brings with it multiple challenges, among them, transactional operations where multiple microservices are involved representing a latent challenge to organizations wishing to adopt a microservice-based architecture (Di Francesco et al., 2019; Knoche & Hasselbring, 2019; Waseem et al., 2020). The above is because in MSA such transactions may involve databases (some even without transactional support) and asynchronous communication elements. The latter with unknown duration (long-lived transactions) which makes it unfeasible to block records, thus making it difficult to know the truth of the system state (Elmagarmid, 1992; Knoche & Hasselbring, 2019).

Considering the above, not all business transactions can comply with ACID properties (Dalal et al., 2003; Gray, 1981; Papazoglou, 2003; Papazoglou & Kratz, 2007; Sun & Aiello, 2007; Tang et al., 2011; Tang et al., 2006), so we can conclude that in transactional operations that span more than one microservice the traditional transactional model is not viable, being necessary to raise alternatives that allow the realization of these operations. It is possible to look for an alternative in the set of Advanced Transactional Models (ATM - Advance Transactional Model) or Extended Transactional Models, which relax the ACID properties, not fulfilling one or some of these properties, to support long-duration transactions (Elmagarmid, 1992; Torres et al., 2009).

This work seeks to adapt an MTA for use in MSA as an alternative for the implementation of transactional operations involving multiple microservices. To achieve this, methodologically, 5 stages are proposed: Statement of the objective, Collection and organization of information, Design, Evaluation, and Publication of results. These stages are based on the general research method referred



to by Sánchez (2004), in addition to elements of the qualitative research workflow described by Hernández-Sampieri and Torres (2018).

Keywords: *advanced transactional models; long-running transactions; microservices; MSA*

1 Introducción

La arquitectura de microservicios (MSA - Microservice Architecture) ha ganado reconocimiento y aceptación en la industria del software, usándose cada vez más en el desarrollo de nuevos sistemas (Alshuqayran et al., 2016; Eismann et al., 2020; Marquez et al., 2018). Existen múltiples razones para adoptar la arquitectura basada en microservicios, siendo las más comunes la escalabilidad y la mantenibilidad (Butzin et al., 2016). Adicionalmente MSA provee beneficios tales como una base de código más pequeña que facilita el desarrollo, prueba e implementación de código más rápidos, optimización de la plataforma según las necesidades del microservicio, soporte para equipos de desarrollo independientes entre otros (Chandramouli, 2019; Hassan et al., 2019).

Sin embargo, la MSA trae consigo múltiples desafíos, entre ellos, las operaciones transaccionales donde participan múltiples microservicios representan un reto latente a las organizaciones que desean adoptar una arquitectura basada en microservicios (Di Francesco et al., 2019; Knoche & Hasselbring, 2019; Waseem et al., 2020). Lo anterior, debido principalmente a que en MSA dichas transacciones pueden involucrar diversas bases de datos, además de elementos de comunicación asíncrona con duración desconocida donde la transacción puede llegar a durar segundos, minutos, horas e incluso semanas (transacciones de larga duración) haciendo inviable el bloqueo de registros y dificultando así conocer la verdad del estado del sistema (Elmagarmid, 1992; Knoche & Hasselbring, 2019).

Con el fin de abordar el reto planteado anteriormente, existen diferentes enfoques: a) los centrados en el diseño, b) los basados en el patrón de diseño SAGAS (Garcia-Molina & Salem, 1987) y c) los basados en el uso del protocolo Two Phase Commit (2PC). Estos enfoques son pocos y todos ellos presentan algún tipo de inconveniente. De ellos el más ampliamente utilizado es SAGAS (Singh & Deshpande, 2020), el cual hace parte de un conjunto de modelo denominado Modelos Transaccionales Avanzados (ATM – Advance Transactional Model) o Modelos Transaccionales Extendidos, los cuales se enfocan en las transacciones de larga duración (Elmagarmid, 1992; Torres et al., 2009). Pese a la existencia de los enfoques de solución mencionados anteriormente, la adopción de la arquitectura basada en microservicios, especialmente en sistemas altamente transaccionales, se ve afectada por la complejidad de implementar operaciones transaccionales que involucran múltiples microservicios (Di Francesco et al., 2019; Knoche & Hasselbring, 2019; Neves, 2019; Viggiano et al., 2018; Waseem et al., 2020).

Este trabajo busca una alternativa para la implementación de transacciones de larga duración en microservicios, por lo que se ha propuesto seleccionar y adaptar un MTA para su uso en MSA. Para lograrlo, metodológicamente se plantea 5 etapas: Planteamiento del objetivo, Reunión y organización de información, Diseño, Evaluación y Presentación de resultados. Estas etapas se basan en el método general de investigación referido por Sánchez (2004), además de elementos del flujo



de trabajo de la investigación cualitativa descritos por Hernández-Sampieri and Torres (2018). Como resultados preliminares de este trabajo se identificaron diferentes MTA, se evaluaron algunas de sus características proyectando una posible adaptación a MSA, así mismo, se identificaron varias mallas de servicio como una alternativa para la implementación del nuevo modelo transaccional para MSA. Adicionalmente, se identificaron modelos de prueba a ser usados durante la evaluación de la solución propuesta.

Este documento está estructurado de la siguiente manera. La sección 2 presenta los trabajos relacionados. La sección 3 describe la metodología usada para el desarrollo del trabajo. La sección 4 presenta los resultados preliminares del trabajo. la sección 5 muestra las conclusiones y la sección 6 las referencias.

2 Trabajos relacionados

La implementación de transacciones que involucran múltiples microservicios es una preocupación al adoptar este tipo de arquitectura (Di Francesco et al., 2019; Knoche & Hasselbring, 2019; Neves, 2019; Vigiato et al., 2018; Waseem et al., 2020). Lo anterior ha generados trabajos que buscan afrontar dicha preocupación. De estos trabajos podemos destacar a Limón et al. (2018) quienes presentaron el framework SagaMAS para el manejo de las transacciones en microservicios, promoviendo un modelado de alto nivel para el manejo de transacciones distribuidas escalables y heterogéneas. Así mismo, en 2019 Zhang et al. (2019) presentaron GRIT, un sistema que usa el protocolo de control de concurrencia optimista para lograr la implementación de transacciones en MSA. Posteriormente, Malyuga et al. (2020) propusieron un modelo basado en el uso del clúster de SAGA y optimizaciones adicionales de su estructura, con el fin de proporcionar una solución tolerante a fallas con características mejoradas de tiempo y memoria para el uso del modelo transaccional SAGAS.

Estos trabajos presentan alternativas para la implementación de transacciones en MSA, sin embargo, ninguna de ellas ha explorado la posibilidad de adaptar un MTA para su uso en MSA.

3 Metodología

La metodología consta de 5 etapas (ver Ilustración 1) adaptadas del método general de investigación referido por Sánchez (2004) e incorporar elementos del flujo de trabajo de la investigación cualitativa descritos por Hernández-Sampieri and Torres (2018).



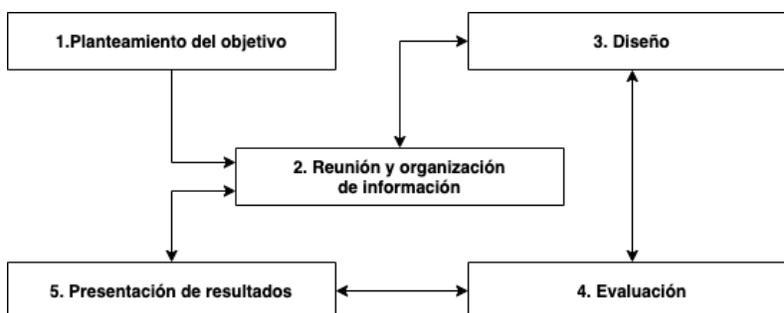


Ilustración 1. Etapas de la metodología

Planteamiento del objetivo: En esta fase se identificó el problema, se definió el objetivo y se establecieron las demás bases de la presente propuesta.

Reunión y organización de información: En esta etapa se identificó los datos y estudios relevantes para la investigación. Para lo cual se han realizado diversos estudios de mapeo sistemático (**SMS** por sus siglas en inglés *Sistematically Mapping Study*), dentro de los cuales se adoptaron prácticas establecidas en ingeniería de software basada en evidencias estudios (Kitchenham et al., 2010), e incorporaron elementos de otros trabajos (Budgen et al., 2008; Mourão et al., 2017; Nguyen et al., 2015; Sepúlveda-Rodríguez et al., 2021), además de usar como apoyo la herramienta software SMS-Builder (Candela-Urbe et al., 2022).

Diseño: En esta etapa se están usando los datos recopilados para diseñar el método de evaluación una posible solución. En este punto se parte del conjunto de MTAs identificados y evaluados para planteará un **Diseño** que incorporará un MTA y las modificaciones necesarias para su uso en MSA, además de concebir el método que se empleará para la evaluación de la solución diseñada.

Evaluación: Esta etapa busca evaluar la solución planteada en la etapa anterior, para lo cual, se evaluará la solución propuesta ciñéndose a los lineamientos establecidos durante el diseño de la evaluación, todo esto con el fin de identificará problemas, retos y posibles mejoras a incorporar en la solución propuesta.

Presentación de resultados: Como parte de esta etapa se han han difundido los resultados del trabajo mediante la elaboración de informes presentados en congresos y revistas, dando a conocer los resultados obtenidos en cada una de las fases con los objetivos planteados dándolos a conocer y promoviendo su adopción.

4 Resultados

Como parte de la etapa de reunión y organización de la información se construyó un SMS denominado “Identificación de Modelos Transaccionales Avanzados para la Arquitectura de Microservicios: Un Estudio de Mapeo Sistemático” (Lugo Cabrera et al., 2021), donde se seleccionaron 120 estudios, a través de los cuales se identificó diversos MTA como se muestra en la Ilustración 2.

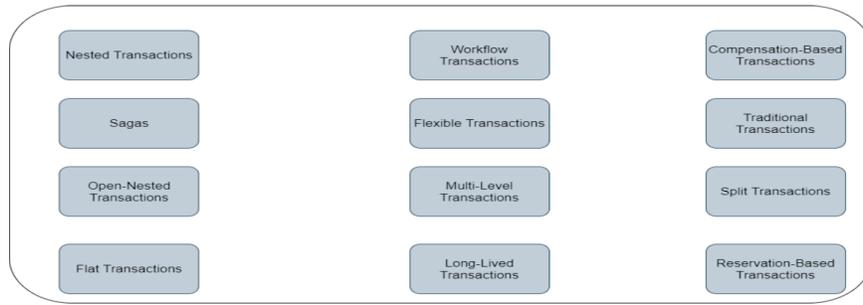


Ilustración 2. Modelos Transaccionales Avanzados identificados (Lugo Cabrera et al., 2021)

Tras la identificación de MTA, se desarrolló un trabajo de evaluación denominado “Modelo transaccional avanzado capaz de adaptarse a las características de la arquitectura basada en microservicios” (Lugo Cabrera et al., 2022). Para la evaluación se usó un proceso formal el cual se dividió en: a) Definición de criterios de evaluación, b) Identificar alternativas de solución, c) Selección escala de evaluación, d) Evaluación y e) Selección. Se analizaron los diferentes modelos con el fin de establecer la viabilidad de su adaptación a MSA usando como criterios el soporte de transacciones de larga duración, la escalabilidad, la complejidad de implementación, el rendimiento, el soporte de transacciones anidadas entre otros. La evaluación postulo el modelo “Nested Transactions”, “Reservation-Based Transaction” y “Flexible Transaction” como los MTA con mayor puntuación como se muestra en la Ilustración 3.

	Compensation based transaction	Flexible Transactions	Multilevel Transactions	Nested Transactions	Open Nested Transactions	Reservation Based Transactions	Sagas	Split Transactions	Workflow Transactions
Escalabilidad	-	Cumple	-	Cumple	Cumple	Cumple-Parcialmente	Cumple	-	-
Complejidad de Implementación	Cumple-Parcialmente	Cumple-Parcialmente	Cumple-Parcialmente	Cumple	Cumple-Parcialmente	Cumple	Cumple	Cumple-Parcialmente	Cumple-Parcialmente
Usabilidad	Cumple-Parcialmente	Cumple-Parcialmente	Cumple-Parcialmente	Cumple-Parcialmente	Cumple-Parcialmente	Cumple-Parcialmente	Cumple-Parcialmente	Cumple-Parcialmente	Cumple-Parcialmente
Rendimiento	Cumple-Parcialmente	Cumple	Cumple	Cumple	Cumple	Cumple	Cumple-Parcialmente	-	Cumple-Parcialmente
Soporte recuperación antes fallos	Cumple	Cumple	Cumple	Cumple	Cumple	Cumple	Cumple-Parcialmente	Cumple-Parcialmente	Cumple-Parcialmente
Soporte transacciones anidadas	Cumple	Cumple	Cumple	Cumple	Cumple-Parcialmente	Cumple	Cumple	Cumple	Cumple
Puntaje Final	13	16	14	17	15	16	15	11	12

Ilustración 3 Evaluación de MTA para su adaptación a MSA (Lugo Cabrera et al., 2022)

Adicionalmente, se inició con el diseño de la evaluación, donde se pretende establecer una línea base de comparación como elemento fundamental de evaluación para las implementaciones que incluyan el MTA adaptado. Para tal fin, se inició con la designación de los Science Gateway como contexto particular para la implementación de un MSA que incorpore un MTA para la implementación de transacciones. Como parte de este proceso se desarrolló un trabajo denominado “Study-



based Systematic Mapping Analysis of Cloud Technologies for Leveraging IT Resource and Service Management: The Case Study of the Science Gateway Approach” (Sepúlveda-Rodríguez et al., 2021), donde se evidenció el éxito de la implementación de dichos sistemas bajo tecnologías de computación en la nube, haciendo adecuado para MSA.

Adicionalmente, se realizaron dos estudios adicionales, uno denominado “Problemas en la Implementación de Pruebas en Sistemas de Microservicios: Estudio de Mapeo Sistemático” (Candela Uribe et al., 2021), donde se identificaron los principales problemas (ver Ilustración 4) relacionados con las pruebas de microservicios y sus posibles causas (ver Ilustración 5), con el fin de poder plantear una estrategia de pruebas que permita disminuir riesgos asociados a la implementación defectuosa del modelo.

El otro trabajo denominado “Service mesh como soporte de transacciones distribuidas en la arquitectura de microservicios.” (Moreno-Cuesta et al., 2022), se enfocó en la identificación varias mallas de servicio como una alternativa para la implementación del nuevo modelo transaccional para MSA.

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

Ilustración 4 Problemas en la implementación de pruebas en MSA (Candela Uribe et al., 2021)

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

Ilustración 5 Posibles causas de Problemas en la implementación de pruebas en MSA (Candela Uribe et al., 2021)

5 Conclusiones

Los SMS han sido fundamentales para el desarrollo del trabajo, permitiendo identificar información y estudios relevantes que se han constituido en la base documental del trabajo. Se identificaron MTA susceptibles de ser adaptados para su uso en MSA, donde se destacaron los modelos "Nested Transactions", "Reservation-Based Transaction" y "Flexible Transaction". Se establecieron los Science Gateway como escenario para la evaluación del MTA adaptado para la implementación de transacciones de larga duración en MSA. Así mismo, se realizó un SMS que permitió identificar los principales problemas relacionados con las pruebas de microservicios y sus posibles causas, con el fin de poder plantear una estrategia de pruebas que permita disminuir riesgos asociados a la implementación defectuosa del modelo. El siguiente paso en el proyecto es determinar la arquitectura para la evaluación del MTA y establecer la línea base comparación, para proceder con la implementación de la solución propuesta y su correspondiente evaluación.

6 Referencias

- Alshuqayran, N., Ali, N., & Evans, R. (2016, 4-6 Nov. 2016). A Systematic Mapping Study in Microservice Architecture. 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA),
- Budgen, D., Turner, M., Brereton, P., & Kitchenham, B. (2008). *Using Mapping Studies in Software Engineering* (Vol. 2). <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>
- Butzin, B., Golasowski, F., & Timmermann, D. (2016). Microservices approach for the internet of things. 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA),
- Candela Uribe, C. A., Gómez-Betancur, D., Sepúlveda-Rodríguez, L., Chavarro-Porrás, J., & Sanabria-Ordoñez, J. (2021). Problemas en la Implementación de Pruebas en Sistemas de Microservicios:



- Estudio de Mapeo Sistemático. *Investigación e Innovación en Ingenierías*, 9(3), 3-17. <https://doi.org/10.17081/invinno.9.3.5314>
- Candela-Uribe, C. A., Sepúlveda-Rodríguez, L. E., Chavarro-Porras, J. C., Sanabria-Ordoñez, J. A., Garrido, J. L., Rodríguez-Domínguez, C., & Guerrero-Contreras, G. (2022). SMS-Builder: An adaptive software tool for building systematic mapping studies. *SoftwareX*, 17, 100935. <https://doi.org/https://doi.org/10.1016/j.softx.2021.100935>
 - Chandramouli, R. (2019). Security Strategies for Microservices-based Application Systems. In (Vol. 800-204): NIST.
 - Dalal, S., Temel, S., Little, M., Potts, M., & Webber, J. (2003). Coordinating business transactions on the web. *IEEE Internet Computing*, 7(1), 30-39.
 - Di Francesco, P., Lago, P., & Malavolta, I. (2019). Architecting with microservices: A systematic mapping study [Article]. *Journal of Systems & Software*, 150, 77-97. <https://doi.org/10.1016/j.jss.2019.01.001>
 - Eismann, S., Bezemer, C. P., Shang, W., Okanović, D., & Van Hoorn, A. (2020, 2020). Microservices: A performance tester's dream or nightmare? ICPE '20 New York, NY, USA.
 - Elmagarmid, A. K. (1992). *Database transaction models for advanced applications*. Morgan Kaufmann Publishers Inc.
 - Garcia-Molina, H., & Salem, K. (1987, 1987). Sagas. SIGMOD '87 New York, NY, USA.
 - Gray, J. (1981). The transaction concept: Virtues and limitations. VLDB,
 - Hassan, S., Bahsoon, R., & Kazman, R. (2019). *Microservice Transition and its Granularity Problem: A Systematic Mapping Study*.
 - Hernández-Sampieri, R., & Torres, C. P. M. (2018). *Metodología de la investigación* (Vol. 4). McGraw-Hill Interamericana México^ eD. F DF.
 - Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering—a tertiary study. *Information and software technology*, 52(8), 792-805.
 - Knoche, H. h. i. u.-k. d., & Hasselbring, W. (2019). Drivers and Barriers for Microservice Adoption - A Survey among Professionals in Germany [Article]. *Enterprise Modelling & Information Systems Architectures*, 14(1), 1-35. <https://doi.org/10.18417/emisa.14.1>
 - Limón, X., Guerra-Hernández, A., Sánchez-García, A. J., & Arriaga, J. C. P. (2018). SagaMAS: a software framework for distributed transactions in the microservice architecture. 2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT),
 - Lugo Cabrera, C. M., Candela-Uribe, C. A., Gómez-Betancur, D., Chavarro-Porras, J. C., Sepúlveda-Rodríguez, L. E., & Sanabria-Ordoñez, J. A. (2021, 2021-11-17). Identificación de Modelos Transaccionales Avanzados para la Arquitectura de Microservicios: Un Estudio de Mapeo Sistemático. 3er Congreso Internacional de TIC para la Amazonía – Retos para la Transformación Digital,
 - Lugo Cabrera, C. M., Candela-Uribe, C. A., Sepúlveda-Rodríguez, L. E., Chavarro-Porras, J. C., & Sanabria-Ordoñez, J. A. (2022). Modelo transaccional avanzado capaz de adaptarse a las características de la arquitectura basada en microservicios. I ENCUESTRO INTERNACIONAL DE EXPERIENCIAS EN INVESTIGACIÓN FORMATIVA - EIEIF2022, II CONGRESO DE INVESTIGACIÓN E INNOVACIÓN EDUCAJAS,
 - Malyuga, K., Perl, O., Slapoguzov, A., & Perl, I. (2020). Fault tolerant central saga orchestrator in RESTful architecture. 2020 26th Conference of Open Innovations Association (FRUCT),
 - Marquez, G., Osses, F., & Astudillo, H. (2018). Review of architectural patterns and tactics for microservices in academic and industrial literature. *IEEE Latin America Transactions*, 16(9), 2321-2327.
 - Moreno-Cuesta, A., Candela-Uribe, C. A., Sepúlveda-Rodríguez, L. E., Chavarro-Porras, J. C., & Sanabria-Ordoñez, J. A. (2022). Service mesh como soporte de transacciones distribuidas en la arquitectura de microservicios. I ENCUESTRO INTERNACIONAL DE EXPERIENCIAS EN INVESTIGACIÓN FORMATIVA - EIEIF2022, II CONGRESO DE INVESTIGACIÓN E INNOVACIÓN EDUCAJAS,



- Mourão, E., Kalinowski, M., Murta, L., Mendes, E., & Wohlin, C. (2017). Investigating the use of a hybrid search strategy for systematic reviews. 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM),
- Neves, J. C. R. D. (2019). *Technical Challenges of Microservices Migration*
- Nguyen, P. H., Kramer, M., Klein, J., & Le Traon, Y. (2015). An extensive systematic review on the model-driven development of secure systems. *Information and software technology*, 68, 62-81.
- Papazoglou, M. P. (2003). Web services and business transactions. *World Wide Web*, 6(1), 49-91.
- Papazoglou, M. P., & Kratz, B. (2007). Web services technology in support of business transactions. *Service Oriented Computing and Applications*, 1(1), 51-63.
- Sánchez, J. C. (2004). *Metodología de la investigación científica y tecnológica*. Ediciones Díaz de Santos.
- Sepúlveda-Rodríguez, L. E., Garrido, J. L., Chavarro-Porras, J. C., Sanabria-Ordoñez, J. A., Candela-Uribe, C. A., Rodríguez-Domínguez, C., & Guerrero-Contreras, G. (2021). Study-based Systematic Mapping Analysis of Cloud Technologies for Leveraging IT Resource and Service Management: The Case Study of the Science Gateway Approach. *Journal of Grid Computing*, 19(4), 41. <https://doi.org/10.1007/s10723-021-09587-7>
- Singh, N. P., & Deshpande, A. (2020). *Solving distributed transaction management problem in microservices architecture using Saga*. ibm.com. Retrieved 2021-10-28 from <https://developer.ibm.com/articles/use-saga-to-solve-distributed-transaction-management-problems-in-a-microservices-architecture/>
- Sun, C., & Aiello, M. (2007). Requirements and evaluation of protocols and tools for transaction management in service centric systems. 31st Annual International Computer Software and Applications Conference (COMPSAC 2007),
- Tang, F., You, I., Li, L., Wang, C.-L., Cheng, Z., & Guo, S. (2011). A pipeline-based approach for long transaction processing in web service environments. *International Journal of Web and Grid Services*, 7(2), 190-207.
- Tang, F.-L., Li, M.-L., & Huang, J. Z. (2006). Automatic transaction compensation for reliable grid applications. *Journal of Computer Science and Technology*, 21(4), 529-536.
- Torres, J., Juárez, E., Doderó, J. M., Aedo, I., & de Yucatan, U. (2009). Advanced transactional models for complex learning processes. *Recursos Digitales para el Aprendizaje*, 360-368.
- Viggiano, M., Terra, R., Rocha, H., Tulio Valente, M., & Figueiredo, E. (2018). Microservices in Practice: A Survey Study. *eprint arXiv:1808.04836*. <http://adsabs.harvard.edu/abs/2018arXiv180804836V>
- Waseem, M., Liang, P., & Shahin, M. (2020). A Systematic Mapping Study on Microservices Architecture in DevOps. *Journal of Systems and Software*, 110798.
- Zhang, G., Ren, K., Ahn, J.-S., & Ben-Romdhane, S. (2019). GRIT: consistent distributed transactions across polyglot microservices with multiple databases. 2019 IEEE 35th International Conference on Data Engineering (ICDE),

Sobre los autores

- **Christian Andrés Candela Uribe:** Ingeniero Sistemas y Computación, Máster en comercio electrónico, Candidato a Doctor en Ingeniería de la Universidad Tecnológica de Pereira. Profesor asociado. christiancandela@uniquindio.edu.co



Los puntos de vista expresados en este artículo no reflejan necesariamente la opinión de la Asociación Colombiana de Facultades de Ingeniería.

Copyright © 2022 Asociación Colombiana de Facultades de Ingeniería (ACOFI)

