

Explorando el Uso de Inteligencia Artificial en el Descubrimiento y Diseño de Microservicios

Daniel Narváez¹

¹CAETI, Facultad de Tecnología Informática – Universidad Abierta Interamericana Argentina

JoseDaniel.NarvaezF@alumnos.uai.edu.ar

Abstract. *Microservice design is a significant challenge in greenfield software development, especially in ensuring internal cohesion and low coupling. This paper proposes an artificial intelligence-based approach, particularly leveraging natural language processing (NLP) and deep learning, to support the discovery and design of microservices from textual requirements. Employing the Design Science Research (DSR) methodology, the proposed research focuses on prompt engineering techniques and architectural validation to guide and assess the quality of microservice-oriented designs. As part of a doctoral project, this work aims to optimize architectural quality, design time, and costs, addressing challenges and opportunities identified through both a systematic mapping study (MSL) and a systematic literature review (SLR).*

Resumen. *El diseño de microservicios es un desafío significativo en desarrollos desde cero, especialmente para garantizar cohesión interna y bajo acoplamiento. Este trabajo propone un enfoque basado en inteligencia artificial, particularmente en procesamiento de lenguaje natural (PLN) y aprendizaje profundo, para apoyar el descubrimiento y diseño de microservicios a partir de requisitos textuales. Utilizando la metodología de Design Science Research (DSR), la propuesta se centra en técnicas de ingeniería de prompts y validación arquitectónica para guiar y evaluar la calidad de diseños orientados a microservicios. Como parte de un proyecto doctoral, esta investigación busca optimizar la calidad arquitectónica, el tiempo de diseño y los costos, abordando retos y oportunidades identificados mediante un mapeo sistemático de la literatura (MSL) y una revisión sistemática de la literatura (SLR).*

Keywords: Ingeniería de Software, Diseño de Software, Microservicios, Inteligencia Artificial, Aprendizaje Automático, Procesamiento de Lenguaje Natural, Investigación en Ciencia del Diseño.

1. Introducción

El diseño de microservicios ha emergido como una práctica clave en el desarrollo de sistemas modernos debido a su capacidad para promover la modularidad, la escalabilidad y la facilidad de mantenimiento en arquitecturas de software complejas [Taibi et al. 2017]. Este paradigma arquitectónico permite dividir aplicaciones monolíticas en componentes independientes y autónomos, denominados microservicios, que interactúan a través de interfaces bien definidas [Newman 2015]. Sin embargo, el proceso de diseño de microservicios sigue siendo un desafío significativo, especialmente

en desarrollos desde cero, donde no existen puntos de referencia arquitectónicos claros [Ait Said et al. 2024].

Esta investigación forma parte de mi proyecto doctoral en el Doctorado en Ciencias Informáticas de la Universidad Abierta Interamericana, el cual aborda el descubrimiento y diseño de microservicios en desarrollos desde cero utilizando requisitos textuales. El objetivo principal de la tesis es proponer una metodología basada en inteligencia artificial (IA), centrada en ingeniería de prompts y validación cuantitativa, para guiar el descubrimiento de microservicios. Este trabajo se alinea con los desafíos y oportunidades explorados en el presente estudio, destacando la importancia de combinar técnicas innovadoras para superar las limitaciones actuales.

Uno de los principales obstáculos es garantizar que los microservicios diseñados cumplan con principios fundamentales, como la cohesión interna y el bajo acoplamiento. Según [Narváez et al. 2024], estos principios son esenciales para maximizar la eficiencia operativa y la flexibilidad del sistema, pero su aplicación práctica depende en gran medida de la experiencia del diseñador y de la calidad de los requisitos disponibles. En entornos donde los requisitos se presentan en forma textual y no estructurada, este desafío se amplifica, lo que aumenta el riesgo de errores de diseño y retrabajo [Taibi et al. 2017].

La integración de IA, particularmente el procesamiento de lenguaje natural (PLN) y el aprendizaje profundo, representa una oportunidad única para abordar estas limitaciones. Estas tecnologías han demostrado su capacidad para extraer información estructurada de textos no estructurados, lo que las hace ideales para analizar documentos de requisitos [Narváez et al. 2024]. Además, estudios recientes destacan la efectividad del clustering y otras técnicas de aprendizaje automático en la identificación de dominios funcionales y módulos coherentes [Ait Said et al. 2024].

Nuestro mapeo sistemático de la literatura (MSL) [Narváez et al. 2024] identificó varias brechas en el uso de IA para el diseño de microservicios. Mientras que algunas investigaciones han explorado el uso de PLN para extraer requisitos, pocas han integrado estas técnicas con herramientas de diseño automatizado que permitan generar microservicios alineados con objetivos arquitectónicos claros. Este vacío teórico y práctico subraya la necesidad de un enfoque más integral que combine la extracción de requisitos textuales con métodos automatizados de diseño y validación.

Además, los desafíos relacionados con la evaluación de la calidad arquitectónica de los microservicios generados, como la medición de cohesión, acoplamiento y alineación con los objetivos del negocio, siguen sin resolverse completamente. Según [Taibi et al. 2017], la falta de herramientas automatizadas para esta evaluación complica la adopción de prácticas de diseño basadas en IA en entornos empresariales.

Por lo tanto, este trabajo propone una metodología experimental que utiliza técnicas de IA para automatizar el descubrimiento y diseño de microservicios desde requisitos textuales. Este enfoque no solo optimiza el tiempo y esfuerzo necesario para el diseño, sino que también mejora la calidad arquitectónica al incorporar métricas objetivas y herramientas de validación automática. La metodología de investigación se basa en Design Science Research (DSR), que ofrece un marco estructurado para la creación y validación de artefactos tecnológicos novedosos [Hevner et al. 2004].

2. Preguntas de investigación

1. ¿Qué técnicas de inteligencia artificial pueden emplearse para identificar y diseñar microservicios a partir de requisitos textuales?
2. ¿Cuáles son los principales desafíos técnicos y metodológicos en la automatización del diseño de microservicios?
3. ¿Cómo se puede evaluar la calidad arquitectónica de los microservicios generados automáticamente?

3. Objetivos

1. Diseñar una metodología basada en ingeniería de prompts para identificar microservicios a partir de requisitos textuales usando IA.
2. Evaluar la calidad arquitectónica mediante métricas como cohesión, acoplamiento, independencia funcional y alineación con los objetivos del negocio.
3. Validar la metodología mediante estudios de caso en entornos controlados y reales.

4. Estado del Arte

El diseño de microservicios ha sido un tema central en la investigación arquitectónica de software en la última década, destacándose como una alternativa escalable y modular frente a las arquitecturas monolíticas y orientadas a servicios (SOA) [Newman 2015, Clements 2002]. A pesar de sus ventajas, la transición a microservicios presenta desafíos significativos, particularmente en entornos de desarrollo desde cero. Entre los problemas más comunes se encuentran la correcta identificación de límites de servicios, la gestión de dependencias, y el aseguramiento de la cohesión interna y bajo acoplamiento entre módulos [Ait Said et al. 2024].

4.1. Enfoques Actuales en el Diseño de Microservicios

Los enfoques tradicionales para el diseño de microservicios se han basado en heurísticas manuales, como la identificación de *bounded contexts* mediante Domain-Driven Design (DDD). Aunque ampliamente aceptadas, estas metodologías dependen de la experiencia del arquitecto y son propensas a errores [Booch et al. 2008]. En organizaciones sin una comprensión clara del dominio, es común diseñar servicios redundantes o con alto acoplamiento, lo que dificulta la escalabilidad y el mantenimiento [Taibi et al. 2017].

Recientemente, se han propuesto técnicas automatizadas para mejorar este proceso. Herramientas como Service Cutter y enfoques de *clustering* han demostrado cierto éxito en la identificación de microservicios a partir de artefactos como bases de datos o código fuente [Ait Said et al. 2024], aunque resultan inadecuadas en escenarios greenfield donde tales artefactos no existen.

Por otro lado, tecnologías emergentes como GreenMicro [Bajaj et al. 2022], SEMGROMI [Vera-Rivera et al. 2023] y GTMicro [Bajaj et al. 2024] han mostrado resultados prometedores en desarrollos desde cero. GreenMicro aplica *clustering* sobre casos de uso y entidades de base de datos, SEMGROMI agrupa historias de usuario mediante PLN para mejorar cohesión y reducir acoplamiento, y GTMicro emplea modelos como BERT para organizar requisitos textuales en microservicios coherentes.

Estas herramientas representan avances significativos para el diseño en escenarios greenfield, aunque enfrentan limitaciones derivadas de la ambigüedad y la calidad de los artefactos de entrada.

4.2. Inteligencia Artificial en el Diseño de Microservicios

La inteligencia artificial ha emergido como una alternativa prometedora frente a las limitaciones de los enfoques tradicionales. En particular, el procesamiento de lenguaje natural (PLN) y el aprendizaje profundo han demostrado eficacia en tareas como la extracción de requisitos y la identificación de patrones en texto no estructurado [Taibi et al. 2017]. Modelos como redes neuronales recurrentes (RNN) y transformers, incluyendo BERT, han sido utilizados con éxito para analizar requisitos textuales, extrayendo entidades y relaciones relevantes para el diseño de software [Narváez et al. 2024].

Herramientas como PF4MD [Li et al. 2023] integran métricas de complejidad y análisis de dependencias para apoyar la descomposición de servicios, mediante visualizaciones que facilitan decisiones arquitectónicas. SEMGROMI y GTMicro aplican PLN y aprendizaje profundo para identificar microservicios, mientras que GreenMicro utiliza clustering para generar estructuras modulares desde etapas tempranas.

Nuestro mapeo sistemático de la literatura [Narváez et al. 2024] identificó que, si bien existen avances en la extracción de requisitos mediante IA, persiste una falta de integración entre estas técnicas y los métodos automatizados de diseño de microservicios. La mayoría de los trabajos se enfocan en etapas aisladas del proceso, como el análisis de dependencias o clustering, sin abordar la transición completa desde requisitos textuales hasta microservicios implementables.

4.3. Brechas y Oportunidades de Investigación

Una de las principales brechas identificadas es la falta de metodologías integradas que combinen técnicas de PLN con aprendizaje profundo y clustering para automatizar el diseño de microservicios en escenarios de desarrollo desde cero. Mientras que estudios previos han explorado la identificación de microservicios en sistemas existentes, pocos trabajos han investigado cómo extraer directamente requisitos textuales y transformarlos en diseños modulares [Ait Said et al. 2024, Taibi et al. 2017].

Además, tecnologías como GreenMicro y SEMGROMI, aunque efectivas, carecen de mecanismos para validar automáticamente la calidad arquitectónica de los microservicios generados. Métricas como cohesión, acoplamiento y alineación con los objetivos del negocio son esenciales para garantizar que los servicios diseñados sean viables tanto técnica como funcionalmente, pero estas métricas rara vez se integran en los enfoques existentes [Bajaj et al. 2022, Vera-Rivera et al. 2023].

Finalmente, la integración de modelos generativos, como los LLMs, representa una oportunidad para automatizar decisiones arquitectónicas y sugerir patrones de diseño basados en conocimiento previo. Sin embargo, su implementación práctica aún requiere mayor investigación y validación empírica para demostrar su efectividad en escenarios reales.

En este contexto, la metodología propuesta busca llenar estas brechas mediante la integración de técnicas avanzadas de IA con herramientas de validación arquitectónica,

optimizando tanto la calidad como la eficiencia del proceso de diseño. Este enfoque tiene el potencial de transformar el diseño de microservicios, estableciendo nuevas bases para la investigación y práctica en este campo.

5. Metodología

Este trabajo adopta la metodología Design Science Research (DSR) como enfoque central, dado su enfoque en el desarrollo de artefactos innovadores y su validación rigurosa en entornos prácticos. Hevner et al. [Hevner et al. 2004] destacan que la DSR se compone de tres ciclos principales: el ciclo de relevancia, el ciclo de rigor y el ciclo central de diseño. Estos ciclos permiten garantizar que el artefacto no solo sea innovador y técnicamente sólido, sino también útil y aplicable en contextos reales.

El ciclo de relevancia comenzó con un análisis sistemático de la literatura para identificar las brechas en el diseño de microservicios desde cero. Este análisis incluyó la revisión de enfoques tradicionales, como el Domain-Driven Design (DDD), que aunque efectivos, dependen en gran medida de la experiencia del arquitecto y de artefactos pre-existentes [Booch et al. 2008]. Además, se evaluaron enfoques emergentes basados en inteligencia artificial, como el uso de modelos de procesamiento de lenguaje natural (PLN) y técnicas de aprendizaje profundo para extraer información relevante de textos no estructurados [Ait Said et al. 2024, Narváez et al. 2024]. Este análisis permitió identificar la falta de integración entre técnicas de PLN y herramientas automatizadas de diseño de microservicios como una brecha crítica.

En el ciclo de diseño, se conceptualizó una metodología experimental que combina ingeniería de prompts, PLN y métricas cuantitativas de calidad arquitectónica. El artefacto generado no es una implementación completa, sino una especificación arquitectónica preliminar que representa microservicios potenciales. Dicha especificación se genera a partir de requisitos textuales mediante tres fases principales: extracción de entidades y relaciones, agrupamiento en servicios mediante algoritmos de clustering, y evaluación estructural basada en métricas. El proceso es parcialmente automatizado, con posibilidad de intervención manual para ajustes finos, y la selección de técnicas específicas se basó en su eficacia documentada en la literatura [Narváez et al. 2024, Ait Said et al. 2024].

Finalmente, en el ciclo de evaluación, se validará la metodología propuesta a través de experimentos en entornos simulados y casos reales. Las métricas clave incluirán cohesión interna, acoplamiento entre servicios y alineación con los objetivos del negocio, siguiendo las mejores prácticas propuestas en trabajos previos [Taibi et al. 2017, Clements 2002]. Además, se realizarán pruebas de estrés para evaluar la escalabilidad y robustez de los microservicios generados.

6. Solución propuesta

Este trabajo propone una metodología que transforma requisitos textuales en una especificación arquitectónica preliminar de microservicios mediante técnicas de inteligencia artificial. El proceso, sujeto a ajuste manual, se organiza en tres etapas interdependientes. La validación considera métricas como cohesión, acoplamiento e independencia funcional, aplicadas sobre representaciones estructurales.

Extracción de requisitos textuales. Este componente emplea modelos de PLN avanzados, como BERT y transformers, para analizar documentos de requisitos y extraer

entidades clave, relaciones y acciones. Por ejemplo, en un dominio de comercio electrónico, puede identificar entidades como “cliente” y “producto”, y relaciones como “realiza pedido” o “gestiona inventario”. La entrada son textos no estructurados, y la salida es una representación semiestructurada de conceptos funcionales. Esta etapa utiliza técnicas de aprendizaje supervisado o no supervisado, en función de la disponibilidad de datos etiquetados.

Diseño automatizado de microservicios. A partir de los conceptos extraídos, se aplica clustering con algoritmos como k-means o DBSCAN para agrupar funcionalidades relacionadas en módulos coherentes que representen microservicios potenciales. El proceso permite ajustar hiperparámetros según el contexto y contempla retroalimentación humana para refinar resultados. La salida es una propuesta estructurada de microservicios lista para validación arquitectónica.

Validación arquitectónica. Esta etapa aplica métricas como cohesión, acoplamiento y alineación con objetivos del negocio para evaluar la calidad del diseño. Se generan informes con indicadores clave y se prevé el uso de herramientas visuales para inspección manual. El enfoque combina automatización con supervisión humana para garantizar viabilidad técnica y organizacional.

La metodología se valida inicialmente en entornos simulados, utilizando conjuntos de datos sintéticos y reales. Los resultados de estas validaciones servirán como base para futuras implementaciones en entornos empresariales, donde se medirá su impacto en términos de tiempo de diseño, costos y calidad arquitectónica.

7. Resultados iniciales

Como resultado inicial, completamos un mapeo sistemático de la literatura (MSL) que identificó las tendencias actuales y las principales brechas en la aplicación de inteligencia artificial (IA) al diseño de microservicios [Narváez et al. 2024]. Este trabajo exploró técnicas de IA relevantes, como el aprendizaje profundo, modelos de decisión de Markov, y herramientas como Mono2Micro y PF4MD, las cuales han demostrado un impacto significativo en el monitoreo, la detección de anomalías y la adaptación dinámica de sistemas basados en microservicios [Kalia et al. 2020, Li et al. 2023, MAGABLEH and ALMIANI23 2019].

El MSL destacó cómo Mono2Micro facilita la transición de aplicaciones monolíticas a microservicios mediante técnicas de clustering temporoespacial [Kalia et al. 2020]. Por otro lado, PF4MD combina marcos conceptuales de análisis de requisitos, como los Problem Frames, con métricas de complejidad para guiar la descomposición de sistemas en microservicios. Esta herramienta se ha utilizado exitosamente para visualizar dependencias, diseñar reglas de descomposición y mejorar la toma de decisiones arquitectónicas [Li et al. 2023].

A pesar de estos avances, el MSL subrayó la necesidad de soluciones integrales para diseñar microservicios desde cero, una área aún poco explorada. Además, identificó desafíos como la falta de datasets estandarizados para evaluar herramientas de diseño automatizado y la ausencia de métricas ampliamente adoptadas para medir la cohesión interna y el bajo acoplamiento en los servicios generados.

Además, el MSL destacó que la validación en escenarios empresariales reales será

crucial para la adopción industrial de estas herramientas. Aunque los experimentos controlados han mostrado resultados prometedores, la implementación práctica enfrenta barreras relacionadas con la interoperabilidad de herramientas y la gestión eficiente de recursos. Estos hallazgos confirman la relevancia del problema abordado y proporcionan una base sólida para el desarrollo de enfoques más avanzados que integren IA en el diseño de microservicios desde la etapa inicial.

Como parte de la evolución de esta línea de investigación, recientemente se publicó un segundo estudio de revisión sistemática en la revista Software (MDPI) [Narváez et al. 2025], que amplía la cobertura del campo incluyendo enfoques emergentes como los modelos generativos (LLMs), el aprendizaje por refuerzo y técnicas de documentación arquitectónica asistida por IA. Este nuevo trabajo analizó 43 estudios entre 2018 y 2024, y reafirmó la utilidad de herramientas como Mono2Micro, SEMGROMI y PF4MD, al tiempo que identificó desafíos persistentes en torno a la consistencia de datos distribuidos, la incorporación de requisitos no funcionales, y la necesidad de validaciones empíricas más robustas. Estos hallazgos refuerzan la necesidad de soluciones integrales como la que se propone en esta investigación doctoral, orientadas a automatizar el diseño de arquitecturas de microservicios desde los requerimientos textuales, garantizando además su validación estructural.

8. Conclusiones y trabajo futuro

Este trabajo propone una metodología basada en inteligencia artificial para el descubrimiento y diseño de microservicios a partir de requisitos textuales, integrando técnicas avanzadas de inteligencia artificial, como procesamiento de lenguaje natural y aprendizaje profundo. La propuesta busca superar las limitaciones de enfoques manuales y heurísticos, enfrentando desafíos clave como la identificación de límites de servicio, la cohesión interna y el bajo acoplamiento.

Los resultados iniciales, sustentados en un mapeo sistemático de literatura y una revisión sistemática más reciente, confirman la pertinencia del problema abordado y evidencian vacíos en la integración entre extracción de requisitos, diseño automatizado y validación estructural. Asimismo, se han identificado métricas fundamentales para evaluar la calidad arquitectónica de los microservicios generados.

La propuesta se orienta a optimizar el esfuerzo de diseño y mejorar la calidad estructural mediante un enfoque híbrido que combina automatización y supervisión humana, ofreciendo una solución escalable para proyectos desarrollados desde cero.

Como trabajo futuro, se plantea la implementación y validación de la metodología en escenarios reales, a través de estudios de caso en distintos dominios. Se evaluará su impacto en términos de eficiencia, calidad arquitectónica y costos. Además, se explorará el uso de herramientas visuales para facilitar la inspección, y se integrarán modelos generativos guiados por prompts junto con mecanismos de validación formal para apoyar decisiones arquitectónicas automatizadas.

Este trabajo pretende contribuir al cuerpo de conocimiento académico y ofrecer soluciones prácticas para la ingeniería de software, consolidando un puente entre los avances en inteligencia artificial y los desafíos actuales del diseño de microservicios.

Referencias

- Ait Said, M., Ezzati, A., Mihi, S., and Belouaddane, L. (2024). Microservices adoption: An industrial inquiry into factors influencing decisions and implementation strategies. *International Journal of Computing and Digital Systems*, 15:2210–142.
- Bajaj, D., Bharti, U., Gupta, I., Gupta, P., and Yadav, A. (2024). Gtmicro—microservice identification approach based on deep nlp transformer model for greenfield developments. *International Journal of Information Technology*, pages 1–11.
- Bajaj, D., Goel, A., and Gupta, S. C. (2022). Greenmicro: identifying microservices from use cases in greenfield development. *IEEE Access*, 10:67008–67018.
- Booch, G., Maksimchuk, R. A., Engle, M. W., Young, B. J., Connallen, J., and Houston, K. A. (2008). Object-oriented analysis and design with applications. *ACM SIGSOFT software engineering notes*, 33(5):29–29.
- Clements, P. C. (2002). Software architecture in practice. *Diss. Software Engineering Institute*.
- Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.
- Kalia, A. K., Xiao, J., Lin, C., Sinha, S., Rofrano, J., Vukovic, M., and Banerjee, D. (2020). Mono2micro: an ai-based toolchain for evolving monolithic enterprise applications to a microservice architecture. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1606–1610.
- Li, Y., Li, Z., Bu, Y., Xiao, H., and Deng, Y. (2023). Pf4md: A microservice decomposition tool combining problem frames. In *2023 IEEE 31st International Requirements Engineering Conference (RE)*, pages 359–360. IEEE.
- MAGABLEH, B. and ALMIANI23, M. (2019). Deep q learning for self adaptive distributed microservices architecture. *IEEE Acess*.
- Narváez, D., Battaglia, N., Fernández, A., and Rossi, G. (2025). Designing microservices using ai: A systematic literature review. *Software*, 4(1):6.
- Narváez, D., Rossi, G. H., and Battaglia, N. (2024). Aplicación de inteligencia artificial en el diseño de microservicios. In *XXX Congreso Argentino de Ciencias de la Computación (CACIC)(La Plata, 7 al 11 de octubre de 2024)*.
- Newman, S. (2015). *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media.
- Taibi, D., Lenarduzzi, V., and Pahl, C. (2017). Processes, motivations, and issues for migrating to microservices architectures: An empirical investigation. *IEEE Cloud Computing*, 4(5):22–32.
- Vera-Rivera, F. H., Cuadros, E. G. P., Perez, B., Astudillo, H., and Gaona, C. (2023). Semgromi—a semantic grouping algorithm to identifying microservices using semantic similarity of user stories. *PeerJ Computer Science*, 9:e1380.