

Efecto en la experiencia práctica y teórica, al seleccionar tecnologías de desarrollo de software

Effect on practical and theoretical experience, when selecting software development technologies

Luis Antonio Llerena Ocaña^{1*}

E-mail: ua.luislllerena@uniandes.edu.ec

ORCID: <https://orcid.org/0000-0001-6440-0167>

Fausto Alberto Viscaino Naranjo¹

E-mail: ua.faustoviscaino@uniandes.edu.ec

ORCID: <https://orcid.org/0000-0003-1760-6992>

Walter Vinicio Culque Toapanta¹

E-mail: ua.walterculque@uniandes.edu.ec

ORCID: <https://orcid.org/0000-0002-3421-2306>

¹Universidad Regional Autónoma de Los Andes, Ambato. Ecuador.

*Autor para correspondencia

Cita sugerida (APA, séptima edición)

Llerena Ocaña, L. A., Viscaino Naranjo, F. A., & Culque Toapanta, W. V. (2024). Efecto en la experiencia práctica y teórica, al seleccionar tecnologías de desarrollo de software. *Revista Científica Cultura, Comunicación y Desarrollo*, 9(3), 182-188. <http://rccd.ucf.edu.cu/index.php/rccd>

RESUMEN

El estudio analizó las percepciones y experiencias de estudiantes y expertos en el uso de frameworks para el desarrollo de software backend. Por ende, el presente estudio se ha enfocado en identificar las ventajas y limitaciones de distintas tecnologías y el impacto en la toma de decisiones académicas y profesionales. Para ello, se empleó una metodología mixta, al combinar encuestas y entrevistas semiestructuradas. Entre los resultados, se ha observado que los estudiantes valoraron la facilidad de aprendizaje y la versatilidad de los frameworks. Mientras que los expertos priorizaron la eficiencia, seguridad y escalabilidad en contextos empresariales. Esta diferencia en las percepciones evidenció una brecha entre el conocimiento teórico y la experiencia práctica. Por otra parte, los estudiantes, con una visión entusiasta, contrastaron con los expertos, quienes aportaron una evaluación crítica basada en la experiencia profesional. En conclusión, se destacó la importancia de integrar enfoques académicos y prácticos para mejorar la formación de futuros desarrolladores y optimizar la toma de decisiones tecnológicas. Esta integración puede guiar ajustes en los programas educativos y fomentar una preparación más adecuada para enfrentar desafíos reales en el desarrollo de software.

Palabras clave:

Facilidad de aprendizaje, Desarrollo backend, Usuario, Tecnologías emergentes.

ABSTRACT

The study analyzed the perceptions and experiences of students and experts in the use of frameworks for backend software development. Finally, this study has focused on identifying the advantages and limitations of different technologies and the impact on academic and professional decision-making. To do this, a mixed methodology was used, combining surveys and semi-structured interviews. Among the results, it has been observed that students valued the ease of learning and the versatility of the frameworks. While the experts prioritized efficiency, security and scalability in business contexts. This difference in perceptions shows a gap between theoretical knowledge and practical experience. On the other hand, the students, with an enthusiastic vision, contrasted with the experts, who provided a critical evaluation based on professional experience. In conclusion, the importance of integrating academic and practical approaches was highlighted to improve the training of future developers and optimize technological decision making. This integration can guide adjustments in educational programs and encourage better preparation to face real challenges in software development.

Keywords:

Ease of learning, Backend development, User, Emerging technologies.

Introducción

El desarrollo de sistemas de información para la web comenzó en los años 90 con la creación de los primeros sitios web, que estaban compuestos principalmente por páginas estáticas. Con el tiempo, la evolución tecnológica permitió la creación de sistemas web dinámicos e interactivos (Parra, et. al., 2024), al dar lugar a los conceptos de front-end y back-end. El backend se consolidó como una capa esencial en el desarrollo de sistemas de información de red para garantizar el funcionamiento eficiente de sitios web y aplicaciones (Rizzardì, et. al., 2024), al gestionar el contenido y la infraestructura del servidor (Bastidas Fuertes, et. al., 2023a). Esta capa abarca la administración de servidores, bases de datos y el procesamiento y almacenamiento de datos, para el correcto funcionamiento de los sistemas de información en red (Sánchez-Delgado, et. al., 2024).

El creciente requerimiento de desarrolladores acentúa la relevancia crítica del rol del backend en el panorama actual de desarrollo. Al mismo tiempo, la constante evolución tecnológica ha creado diversas oportunidades para satisfacer la demanda de sistemas de red escalables, eficientes y seguros (Alfadel, et. al., 2022). Las tecnologías actuales ofrecen recursos únicos que mejoran el desarrollo del backend. Seleccionar el framework adecuado para un proyecto específico requiere una estrategia y planificación cuidadosa, alineada con las tendencias del mercado.

En 2023, Stack Overflow encuestó a más de 90,000 desarrolladores para determinar las tecnologías más buscadas, utilizadas y populares. Los resultados mostraron que JavaScript ha sido el lenguaje de programación más utilizado durante una década consecutiva, con Python y C# también entre los diez primeros. Además, frameworks como Django (Python), ASP.NET Core (C#) y Node.js (JavaScript) se encuentran entre los más utilizados (Llerena, et. al., 2022), al constituir Node.js el framework backend más empleado por los profesionales (Bastidas Fuertes, et. al., 2023b) (Molina, & Pedreira, 2020) (Soto de la Cruz, et. al., 2023).

Debido a la diversidad de tecnologías disponibles en el entorno digital, el presente estudio se enfoca en analizar y comparar las percepciones y experiencias de estudiantes y expertos en el uso de frameworks para el desarrollo de software backend. Además, se busca identificar las ventajas y limitaciones de diferentes tecnologías y su impacto en la toma de decisiones en el contexto académico y profesional (Llerena, et. al., 2020). Para ello, se identifican los siguientes objetivos específicos:

- Evaluar las percepciones y actitudes de los estudiantes y expertos hacia diferentes frameworks de desarrollo backend.
- Comparar las experiencias y valoraciones de estudiantes y expertos sobre los frameworks de backend, al destacar la brecha entre el conocimiento teórico y la experiencia práctica.
- Proponer soluciones para integrar enfoques académicos y prácticos en la selección y uso de tecnologías de backend.

Materiales y métodos

Este estudio investigó el impacto de los frameworks en el desarrollo de backend, al enfocarse en la percepción y experiencia de estudiantes de Ingeniería de Software y profesionales del Departamento de Desarrollo de la UNIANDÉS en Ecuador. La muestra incluyó 87 estudiantes que utilizan Node.js y Django, así como 7 ingenieros de software que trabajan con .NET.

Para la recolección de datos, se empleó una metodología mixta que combinó datos cualitativos y cuantitativos (Granikov, et. al., 2020). Se distribuyeron encuestas electrónicas para recopilar información sobre el uso de frameworks, la satisfacción del usuario y la percepción de la eficacia en el desarrollo de backend. Paralelamente, se realizaron entrevistas semiestructuradas a una muestra representativa para explorar en profundidad la experiencia con los frameworks, al identificar los desafíos y obtener una información cualitativa adicional (Zhang, et. al., 2023).

Los datos cuantitativos obtenidos de las encuestas se analizaron descriptivamente para identificar tendencias y patrones. Por otro lado, los datos cualitativos de las entrevistas se transcribieron y analizaron mediante técnicas de análisis de contenido para descubrir temas emergentes y patrones en las respuestas (Tramullas, 2020). Para asegurar la integridad del estudio y el cumplimiento de las normas éticas, se siguieron las directrices establecidas por la UNIANDÉS. Se garantizó el anonimato y la confidencialidad de los participantes, quienes proporcionaron su consentimiento informado antes de participar en el estudio.

Resultados-discusión

Percepciones y actitudes de los estudiantes hacia frameworks de desarrollo backend

La facilidad de aprendizaje es crucial para los estudiantes al evaluar frameworks de desarrollo backend. La simplicidad de la sintaxis, la calidad de la documentación y la disponibilidad de recursos educativos son factores determinantes en esta percepción. Frameworks como Node.js y Django se destacan por su curva de aprendizaje accesible, facilitada por abundantes tutoriales y guías. Esta facilidad se incrementa con la sencilla integración y rápida puesta en marcha de proyectos, además del apoyo de documentación y comunidades activas. Incluso, la versatilidad es valorada, de modo que los estudiantes buscan frameworks, al adaptarse a diversos tipos de proyectos y necesidades.

En cambio, Node.js es apreciado por su manejo de aplicaciones en tiempo real y su adopción en el desarrollo web. Por otro lado, Django se destaca por su robustez en aplicaciones web complejas e integración con Python. Sin embargo, los estudiantes prefieren frameworks que permiten experimentar con distintos proyectos y que se adaptan a múltiples escenarios de desarrollo. De modo que se valora por la flexibilidad y la capacidad de integración con otras tecnologías.

La satisfacción general depende de la facilidad de aprendizaje, la versatilidad y el rendimiento durante el desarrollo. Los estudiantes reportan alta satisfacción con frameworks

que cumplen sus expectativas de eficiencia y facilidad de uso. Frameworks como Django REST y Node.js reciben comentarios positivos por su robustez y la experiencia fluida en el desarrollo de aplicaciones. Esta satisfacción se incrementa con un buen rendimiento, estabilidad y el apoyo de comunidades activas y bases de usuarios amplias. Esto facilita la resolución de problemas y la obtención de ayuda.

Opiniones y experiencias de expertos en el uso de frameworks de backend

La eficiencia de un framework es valorada por los expertos, por la capacidad de acelerar el desarrollo, mantener un alto rendimiento en producción y gestionar eficientemente los recursos. No obstante, Node.js destaca por su eficiencia en operaciones de entrada/salida, gracias a su arquitectura basada en eventos y ejecución asíncrona, ideal para aplicaciones en tiempo real y de alta carga. Por otro lado, Django es apreciado por su desarrollo rápido y herramientas integradas, al facilitar la creación de aplicaciones robustas. Sin embargo, .NET es reconocido por la robustez y optimización en entornos empresariales complejos, al ofrecer un alto rendimiento en sistemas grandes.

En términos de seguridad, los expertos buscan frameworks con mecanismos sólidos contra vulnerabilidades y ataques. En cambio, Node.js requiere prácticas rigurosas y revisión continua del código debido a su dependencia de implementaciones y dependencias. Por el contrario, Django es valorado por su protección contra vulnerabilidades comunes como CSRF y XSS, gracias al enfoque en la seguridad desde el diseño. De igual modo, .NET ofrece un amplio conjunto de características de seguridad integradas. De modo que incluye autenticación y autorización robustas, y soporte para políticas de seguridad a nivel de plataforma, especialmente en integración con los servicios de seguridad de Microsoft.

La escalabilidad constituye un factor determinante para los expertos, quienes buscan frameworks capaces de soportar

el crecimiento de las aplicaciones y adaptarse a cambios en la demanda sin comprometer el rendimiento. En cambio, Node.js sobresale por su modelo de ejecución asíncrono y manejo de múltiples conexiones simultáneas, al ser altamente escalable para aplicaciones de red y servicios en tiempo real. Al contrario de Django, aunque escalable, puede enfrentar desafíos en aplicaciones extremadamente grandes sin estrategias de optimización adecuadas. Por otro lado, .NET ofrece excelentes capacidades de escalabilidad, especialmente con servicios en la nube como Azure, gracias a su arquitectura modular y soporte para microservicios.

En último lugar, al hablar de los desafíos existentes, se constata que en entornos empresariales coexisten los problemas de implementación, mantenimiento y gestión de recursos. Entre ellos, se observa a Node.js que enfrenta dificultades con la gestión de dependencias y compatibilidad entre versiones. Además, incide en falta de soluciones estandarizadas para la gestión de procesos y monitoreo en producción. De modo similar, Django presenta retos en la configuración de entornos de producción y optimización para grandes volúmenes de datos, con la inclusión de una curva de aprendizaje pronunciada. Finalmente, NET, aunque potente, presenta complejidades en la configuración y mantenimiento de aplicaciones a gran escala, junto con costos asociados a licencias y la necesidad de compatibilidad con la infraestructura de Microsoft.

Comparación entre estudiantes y expertos sobre la tecnología de backend

A continuación, se presentan las diferencias en la valoración de frameworks entre estudiantes y expertos. La siguiente tabla analiza cómo cada grupo prioriza diferentes aspectos, como la facilidad de aprendizaje, la versatilidad, la satisfacción general y el soporte técnico, al reflejar las respectivas necesidades y enfoques en el desarrollo de software (ver tabla 1).

Tabla 1: Valoraciones de estudiantes vs. Expertos respecto a frameworks.

Aspecto	Enfoque	Estudiantes	Expertos
Facilidad de aprendizaje vs. Experiencia práctica.	Valoración	Los estudiantes valoran la facilidad de aprendizaje y la accesibilidad. Buscan herramientas intuitivas con una curva de aprendizaje suave y documentación extensa.	Los expertos valoran la capacidad del framework para enfrentar desafíos complejos y sus características avanzadas. Prefieren herramientas robustas y flexibles para entornos de producción.
	Razones subyacentes	La facilidad de aprendizaje es prioritaria para minimizar la complejidad inicial y enfocarse en conceptos básicos del desarrollo de software.	La experiencia práctica y la necesidad de resolver problemas reales hacen que la facilidad de aprendizaje sea menos necesaria. Se priorizan aspectos técnicos avanzados y la capacidad de integración y escalabilidad.
Versatilidad vs. Eficiencia en escenarios reales.	Valoración	Los estudiantes valoran la versatilidad para experimentar con diversos tipos de proyectos y explorar diferentes enfoques de desarrollo.	Los expertos valoran la eficiencia en términos de rendimiento y manejo de carga. La capacidad para gestionar grandes volúmenes de datos y solicitudes es crucial.
	Razones subyacentes	La versatilidad permite a los estudiantes desarrollar una gama diversa de habilidades y obtener una comprensión más amplia del desarrollo backend.	La eficiencia en escenarios reales es fundamental para asegurar la estabilidad y el rendimiento de las aplicaciones en producción. Los expertos buscan soluciones efectivas para desafíos técnicos específicos.

Satisfacción general vs. desafíos técnicos.	Valoración	La satisfacción general se basa en la experiencia de aprendizaje y la facilidad de uso. Los estudiantes valoran herramientas que faciliten el desarrollo y ofrezcan una experiencia amigable.	La satisfacción se basa en cómo el framework maneja desafíos técnicos y limitaciones en entornos empresariales. La capacidad para superar obstáculos técnicos es clave.
	Razones subyacentes	La facilidad para aprender y usar el framework impulsa el entusiasmo y la motivación en el proceso de aprendizaje.	Los desafíos técnicos y las limitaciones prácticas influyen en la percepción de los expertos. La habilidad del framework para ofrecer soporte robusto en situaciones reales determina su satisfacción.
Documentación y soporte vs. integración y compatibilidad	Valoración	Los estudiantes valoran la documentación clara y el soporte accesible como esenciales para la formación.	Los expertos valoran la integración y compatibilidad del framework con otras herramientas y tecnologías. La capacidad de integrarse con sistemas existentes es fundamental.
	Razones subyacentes	La documentación y el soporte son cruciales para superar dificultades durante el aprendizaje y proporcionar una guía esencial para el desarrollo de proyectos.	La integración con otras tecnologías y la compatibilidad con sistemas existentes son críticas para el éxito de proyectos complejos en entornos empresariales.

Fuente: Elaboración propia.

Brecha entre conocimiento teórico y experiencia práctica

Las diferencias entre estudiantes y expertos en el uso de frameworks backend son marcadas y significativas. Los estudiantes se enfocan en conceptos básicos y teorías en un entorno académico, con poca experiencia práctica. Mientras que los expertos aplican su conocimiento en producción, al enfrentar desafíos como la integración de sistemas y la optimización del rendimiento. Incluso obtienen una comprensión profunda de las tecnologías.

A pesar de ello, los estudiantes eligen frameworks por su facilidad de aprendizaje y popularidad, mientras que los expertos los seleccionan por criterios técnicos como escalabilidad y eficiencia. En cuanto a la implementación, los expertos se centran en la optimización y seguridad, al detectar problemas que los estudiantes podrían pasar por alto.

Finalmente, en términos de innovación, los estudiantes adoptan nuevas tecnologías con entusiasmo, pero a veces sobreestiman sus capacidades. Mientras, los expertos, en cambio, buscan mejorar y adaptar tecnologías existentes para resolver problemas reales.

Propuesta de integración académica y práctica en tecnologías de backend

Para mejorar la integración de enfoques académicos y prácticos en la selección de tecnologías de backend, se proponen diversas soluciones que abarcan desde la incorporación de experiencias prácticas en el currículo hasta la promoción de la colaboración entre la academia y la industria. La siguiente tabla detalla las acciones propuestas, la descripción y los beneficios esperados, al proporcionar una guía clara para implementar estas mejoras (ver tabla 2).

Tabla 2: Acciones para integrar enfoques académicos y prácticos en tecnologías de backend.

Área de mejora	Acción propuesta	Descripción	Beneficios
1. Desarrollo de programas de formación integrada.	1.1. Incorporación de experiencias prácticas.	Simulaciones de proyectos reales: Implementar proyectos que reflejen escenarios del mundo real en el currículo académico.	Experiencia práctica realista y comprensión más profunda de los desafíos.
		Colaboraciones con empresas: establecer asociaciones con empresas para ofrecer prácticas y proyectos colaborativos.	
	1.2. Incorporación de tecnología y herramientas actuales.	Actualización continua del contenido: Asegurar que los contenidos del curso se actualicen regularmente para reflejar las últimas tecnologías y prácticas del mercado.	Preparación para un entorno tecnológico en constante cambio.
		Capacitación en nuevas tecnologías: Ofrecer módulos adicionales o cursos electivos sobre tecnologías emergentes y tendencias actuales.	

2. Promoción de la colaboración entre academia e industria.	2.1. Creación de redes de mentoring y tutoring.	Mentoring profesional: Establecer programas de mentoría donde los expertos en la industria guíen a los estudiantes a través de proyectos y desafíos reales.	Perspectiva práctica y consejos basados en la experiencia profesional.
		Seminarios y talleres: organizar seminarios y talleres donde los expertos compartan sus experiencias y conocimientos sobre la selección y uso de tecnologías de backend.	
	2.2. Proyectos conjuntos y estudio de casos.	Proyectos de colaboración: Facilitar proyectos conjuntos entre estudiantes y profesionales. De modo que permita a los estudiantes aplicar los conocimientos teóricos mientras reciben retroalimentación práctica.	Aplicación práctica del conocimiento y retroalimentación directa.
		Estudio de casos reales: integrar estudios de casos basados en proyectos reales en el currículo académico.	
3. Fomento de la investigación aplicada y la innovación.	3.1. Proyectos de investigación conjunta.	Investigación aplicada: incentivar la investigación aplicada en colaboración entre estudiantes y profesionales para explorar nuevas tecnologías y desarrollar soluciones innovadoras.	Desarrollo de soluciones innovadoras y aplicación práctica.
		Competencias y desafíos: organizar competencias y desafíos donde los estudiantes puedan presentar soluciones innovadoras para problemas reales del sector.	
	3.2. Desarrollo de centros de innovación y laboratorios.	Centros de innovación: crear centros de innovación y laboratorios en universidades donde estudiantes y expertos trabajen juntos en el desarrollo y evaluación de nuevas tecnologías y frameworks.	Evaluación y desarrollo de nuevas tecnologías en entornos controlados.
		Entornos de prueba: establecer entornos de prueba para experimentar con diferentes tecnologías y frameworks, al permitir a los estudiantes y profesionales analizar el rendimiento y la aplicabilidad.	
4. Evaluación continua y retroalimentación.	4.1. Sistemas de retroalimentación bidireccional.	Evaluación de proyectos: implementar sistemas de evaluación donde los expertos proporcionen retroalimentación a los estudiantes sobre sus proyectos y enfoques tecnológicos.	Mejorar habilidades prácticas y ajustar contenido académico a las necesidades del mercado.
		Encuestas y revisiones: realizar encuestas periódicas y revisiones de los currículos académicos en colaboración con profesionales del sector para identificar áreas de mejora.	
	4.2. Adaptación de metodologías de enseñanza.	Metodologías activas: adoptar metodologías activas de enseñanza que incluyan estudios de caso, simulaciones y aprendizaje basado en proyectos.	Experiencia de aprendizaje más cercana a la práctica profesional.
		Aprendizaje basado en problemas: Implementar el aprendizaje basado en problemas (ABP) para que los estudiantes aborden desafíos del mundo real y apliquen sus conocimientos teóricos.	

Fuente: Elaboración propia.

Integrar enfoques académicos y prácticos en la selección de tecnologías de backend requiere colaboración efectiva entre el entorno académico y la industria. Implementar programas de formación integrada, fomentar la colaboración, promover la investigación aplicada y establecer sistemas de retroalimentación continua son pasos clave. Esto optimiza la formación académica y mejora la toma de decisiones en el ámbito profesional. Además, esta integración enriquecería la experiencia educativa de los estudiantes y los prepararía para enfrentar desafíos del desarrollo de software en un entorno empresarial real.

Reflexiones finales

Las actitudes positivas de los estudiantes hacia frameworks como Node.js y Python reflejan la preferencia por tecnologías accesibles y variables. Sin embargo, este enfoque se limita al no abordar desafíos técnicos en entornos profesionales. En cambio, los expertos, por su parte, valoran la eficiencia, seguridad y robustez, cruciales en aplicaciones empresariales. Por lo que conduce a la discrepancia entre estudiantes y expertos al insistir en la necesidad de integrar ambos enfoques para una evaluación completa. Para ello, se debe mejorar la formación académica para incluir aspectos técnicos y prácticos, así como adaptar herramientas emergentes a requisitos profesionales, para facilitar una transición más fluida al entorno laboral.

Conclusiones

El análisis ha mostrado diferencias claras entre estudiantes y expertos en cuanto a frameworks de backend. Los estudiantes valoran la facilidad de aprendizaje y la versatilidad, mientras que los expertos priorizan la eficiencia, escalabilidad y los desafíos de implementación. Esta brecha sugiere que combinar ambas perspectivas podría mejorar la selección y uso de tecnologías en el desarrollo de software.

Los resultados han identificado la necesidad de integrar mejor la teoría académica con la práctica profesional. Incluir escenarios prácticos y desafíos reales en el currículo académico prepararía mejor a los estudiantes para enfrentar los requisitos técnicos del entorno profesional. De modo que mejora así su formación y la toma de decisiones tecnológicas.

Las futuras investigaciones deberían explorar metodologías pedagógicas que incorporen más experiencia práctica, al evaluar cómo estas adaptaciones impactan en el desempeño laboral. También sería útil investigar cómo ajustar tecnologías emergentes para resolver desafíos específicos, lo que contribuiría a crear soluciones más efectivas en el desarrollo de software backend.

Referencias bibliográficas

Alfadel, M., Costa, D. E., Shihab, E., & Adams, B. (2022). On the Discoverability of npm Vulnerabilities in Node.js Projects. *ACM Transactions on Software Engineering and Methodology*, 32(4), 1-27. <https://dl.acm.org/doi/abs/10.1145/3571848>.

Bastidas Fuertes, A., Pérez, M., & Meza, J. (2023b). Morph Framework: An Innovative Approach to Transpiler-Based Multi-Language Software Development. *IEEE Access*, 11(November), 124386-124429. <https://ieeexplore.ieee.org/abstract/document/10309301>.

Bastidas Fuertes, A., Verdú Pérez, M. J., & Meza Hormaza, J. (2023a). Transpiler-Based Architecture Design Model for Back-End Layers in Software Development. *Applied Sciences*, 13(20), 11371. <https://www.mdpi.com/2076-3417/13/20/11371>.

Granikov, V., Hong, Q. N., Crist, E., & Pluye, P. (2020). Mixed methods research in library and information science: A methodological review. *Library & Information Science Research*, 42(1), 3-6. <https://www.sciencedirect.com/science/article/abs/pii/S0740818819302294>

Llerena Ocaña, L. A., González Hernández, W., Llerena Ocaña, L. A., & González Hernández, W. (2020). Formación de la competencia «desarrollar sistemas web en los espacios virtuales de aprendizaje». *Revista Cubana de Educación Superior*, 39(1), 2-12. http://scielo.sld.cu/scielo.php?pid=S0257-43142020000100016&script=sci_arttext.

Llerena Ocaña, L. A., Viscaino Naranjo, F. A., Culque Toapanta, W. V., Llerena Ocaña, L. A., Viscaino Naranjo, F. A., & Culque Toapanta, W. V. (2022). Desarrollo de software con Net Core. *Revista Universidad y Sociedad*, 14(2), 85-89. http://scielo.sld.cu/scielo.php?pid=S2218-36202022000200085&script=sci_arttext

Molina Ríos, J., & Pedreira Souto, N. (2020). Comparison of development methodologies in web applications. *Information and Software Technology*, 119(March), 1-5. <https://www.sciencedirect.com/science/article/abs/pii/S0950584919302551>.

Parra Ullauri, J. M., Madhukumar, H., Nicolaescu, A. C., Zhang, X., Bravalheri, A., Hussain, R., Vasilakos, X., Nejabati, R., & Simeonidou, D. (2024). KubeFlower: A privacy-preserving framework for Kubernetes-based federated learning in cloud-edge environments. *Future generation computer systems*, 157(August), 558-572. <https://www.sciencedirect.com/science/article/pii/S0167739X24001134>.

Rizzardi, A., Sicari, S., & Coen Porisini, A. (2024). Towards rapid modeling and prototyping of indoor and outdoor monitoring applications. *Sustainable computing*, 41(January), 2-9. <https://www.sciencedirect.com/science/article/pii/S2210537923001063>.

Sánchez-Delgado, A., Garg, K., Scherjon, C., & Lee, H. (2024). Frontend and backend electronics achieving flexibility and scalability for tomographic tactile sensing. *Intelligent service robotics*, 17(January), 75-83. <https://link.springer.com/article/10.1007/s11370-023-00502-5>.

Soto de la Cruz, R., Castro Espinoza, F. A., Soto, L., Soto de la Cruz, R., Castro Espinoza, F. A., & Soto, L. (2023). Isodata Based Method for Clustering Surveys Responses with Mixed Data: The 2021 StackOverflow Developer Survey. *Computación y Sistemas*, 27(1), 173-182. https://www.scielo.org.mx/scielo.php?pid=S1405-55462023000100173&script=sci_arttext&tlng=en.

Tramullas, J. (2020). Temas y métodos de investigación en Ciencia de la Información, 2000-2019. Revisión bibliográfica. *El profesional de la información*, 29(4), 2-6. <https://revista.profesionaldelainformacion.com/index.php/EPI/article/view/77328>.

Zhang, C., Tian, L., & Chu, H. (2023). Usage frequency and application variety of research methods in library and information science: Continuous investigation from 1991 to 2021. *Information Processing and Management*, 60(6), 4-8. <https://www.sciencedirect.com/science/article/abs/pii/S0306457323002443>.