

TENDENCIAS EN LENGUAJES DE PROGRAMACIÓN: UN ANÁLISIS BASADO EN OFERTAS LABORALES Y REVISIÓN SISTEMÁTICA

PROGRAMMING LANGUAGE TRENDS: A REVIEW OF JOB MARKET AND SYSTEMATIC REVIEW

Javier Hernán Jiménez Beltrán

Docente de tiempo completo. Universidad Nacional Abierta y a Distancia (UNAD). Grupo Byte in design, Tecnología en Desarrollo de *Software*. <https://orcid.org/0000-0002-2851-8173>
javier.jimenez@unad.edu.co

María Patricia Amórtégui Vargas

Docente de tiempo completo. Universidad Nacional Abierta y a Distancia (UNAD). Grupo Byte in design, Tecnología en Desarrollo de *Software*. <https://orcid.org/0000-0002-6705-9184>
maria.amortegui@unad.edu.co

María Consuelo Rodríguez Niño

Docente de tiempo completo. Universidad Nacional Abierta y a Distancia (UNAD). Grupo Byte in design, Tecnología en Desarrollo de *Software*. <https://orcid.org/0000-0002-4572-3187>
mariac.rodriguez@unad.edu.co

José Miguel Herrán Suárez

Líder del Programa Tecnología en Desarrollo de *Software*. Universidad Nacional Abierta y a Distancia (UNAD). Grupo Byte in design, Tecnología en Desarrollo de *Software*. <https://orcid.org/0000-0002-7596-1397>. jose.herran@unad.edu.co



RESUMEN

Este estudio investiga las tendencias más recientes en lenguajes de programación y bases de datos, un campo en constante evolución. La investigación se realizó mediante un enfoque metodológico mixto, en el cual, en primer lugar, se llevó a cabo una revisión sistemática de la literatura académica reciente, lo que permitió establecer una base teórica para comprender las innovaciones tecnológicas en el desarrollo de *software*. Por otro lado, se emplearon técnicas de *web scraping* para extraer y analizar un amplio conjunto de ofertas laborales publicadas en plataformas digitales en Colombia.

El objetivo principal de esta doble aproximación fue identificar qué lenguajes, *frameworks* y tecnologías son los más demandados por el mercado laboral en la actualidad. Los resultados mostraron hallazgos significativos que indican una alta prevalencia de lenguajes como Python y JavaScript, así como Java, cada uno con un rol predominante en diferentes ámbitos de

desarrollo. En cuanto a las bases de datos, se observó una fuerte demanda tanto de tecnologías SQL como NoSQL, lo que destaca su complementariedad en el mundo real. También se puso en evidencia la relevancia de *frameworks* y librerías clave como React, Angular y Spring.

Estos hallazgos permiten generar recomendaciones prácticas para las instituciones educativas con el fin de alinear sus currículos de los programas de desarrollo de *software* con las necesidades reales de la industria. Así, se busca que los futuros profesionales adquieran las habilidades técnicas más relevantes, o que contribuirá a mejorar su inserción y competitividad en el mercado laboral.

Palabras clave: desarrollo de *software*; programación; lenguajes de programación; *frameworks*; Python; Java.



ABSTRACT

This study investigates the most recent trends in programming languages and databases, a constantly evolving field. The research was carried out through a mixed methodological approach, in which, first, a systematic review of recent academic literature was carried out, which allowed establishing a theoretical basis to understand technological innovations in software development. On the other hand, web scraping techniques were used to extract and analyze a wide set of job offers published on digital platforms in Colombia.

The main objective of this dual approach was to identify which languages, frameworks, and technologies are most in-demand in the current job market. The results yielded significant findings, indicating a high prevalence of languages such as Python and JavaScript, as well as Java, each with a predominant role in different development

domains. In databases, a strong demand for both SQL and NoSQL technologies was observed, highlighting their complementarity in real-world applications. The study also confirmed the relevance of key frameworks and libraries like React, Angular, and Spring.

These findings provide practical recommendations for educational institutions, allowing them to align their Software Development curricula with the real needs of industry. The ultimate goal is to ensure that future professionals acquire the most relevant technical skills, thereby enhancing their employability and competitiveness in the labor market.

Keywords: Software Development; computer programming; programming languages; frameworks; Python; Java



INTRODUCCIÓN

La programación es un área relevante en la industria, debido a que el *software* permite a las empresas optimizar procesos, reducir costos, incorporar innovaciones y mejorar la eficiencia en diferentes sectores. Por esta razón, es necesario contemplar, en la enseñanza de los programas académicos asociados al desarrollo de *software*, las tendencias del mercado para formar profesionales competentes, actualizados y capacitados para dar respuesta a las necesidades del mercado laboral.

En este sentido, corresponde a la academia identificar cuáles son las necesidades del sector productivo en cuanto a aspectos

técnicos de programación, con miras a ofrecer formación pertinente y de calidad y favorecer la actualización constante de sus contenidos. En este contexto, el programa Tecnología en Desarrollo de *Software* de la UNAD realizó una investigación mixta que combinó una revisión de literatura y una búsqueda de ofertas laborales para el área de desarrollo de *software*, con miras a identificar convergencias en las áreas de desarrollo y bases de datos y, así, contrastar los contenidos microcurriculares y realizar las actualizaciones pertinentes. En el siguiente *working paper* se detalla dicha investigación y se presentan los resultados más relevantes.

METODOLOGÍA

La presente investigación se desarrolló mediante una revisión sistemática de literatura (RSL) combinada con un análisis de ofertas laborales. En la revisión de la literatura, se buscaron artículos sobre el uso de lenguajes de programación, *frameworks* y bibliotecas, con el propósito de identificar las tendencias actuales en los lenguajes de

programación y en los sistemas de gestión de bases de datos, con énfasis en su impacto en la industria del *software*. El proceso metodológico se estructuró siguiendo el marco de referencia PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*), lo cual permitió garantizar la transparencia, exhaustividad y reproducibilidad del estudio.

Planeación de la revisión

Durante la fase de planeación, se definió la pregunta de investigación utilizando el enfoque PICOC (*Population, Intervention, Comparison, Outcome, Context*), que permitió delimitar el objeto de estudio de la siguiente manera:

Población: lenguajes de programación y sistemas de bases de datos relacionales (SQL) y no relacionales (NoSQL).

- Intervención: desarrollo de *software*.



- Comparación: prácticas y tendencias en la industria del *software*.
- Resultado esperado (*outcome*): *frameworks* y bibliotecas.
- Contexto: publicaciones académicas y científicas publicadas a partir del 2022.

Con base en esta formulación, la pregunta de investigación central fue: “¿cuáles son las tendencias en lenguajes de programación y sistemas de gestión de bases de datos en la industria del *software*?”

Estrategia de búsqueda

Para la identificación de estudios relevantes, se diseñó una cadena de búsqueda con términos clave y sinónimos, combinados mediante operadores booleanos:

“NoSQL databases” OR “SQL databases” OR “programming languages”) AND (“software development”) AND (“software industry”) AND (“trends”) Las bases de datos consultadas fueron:

- ACM Digital Library
- Google Scholar

- IEEE Xplore
- ISI Web of Science
- ScienceDirect
- Scopus

La búsqueda se limitó a publicaciones comprendidas entre enero del 2022 y diciembre del 2024, con el fin de asegurar la pertinencia y actualidad de la información.

Criterios de inclusión y exclusión

Se establecieron los siguientes criterios para la selección de los estudios:

- Criterios de inclusión:
 - Publicaciones que mencionen explícitamente lenguajes de programación.
 - Publicaciones que aborden bases de datos SQL o NoSQL.
- Criterios de exclusión:
 - Documentos publicados antes del 2022.
 - Estudios que no traten la temática de interés.

Proceso de selección de estudios

Se realizó un proceso en tres etapas:



- **Identificación:** se recopilaron inicialmente 245 publicaciones provenientes de las seis bases de datos consultadas.
- **Depuración:** se eliminaron dos duplicados, por lo que quedaron 243 registros únicos.
- **Elegibilidad y selección final:** tras la aplicación de los criterios de inclusión y exclusión, se aceptaron 25 artículos, mientras que 218 fueron descartados por falta de pertinencia.

Evaluación de calidad

Para garantizar la validez y confiabilidad de los resultados, los estudios seleccionados fueron sometidos a un proceso de evaluación mediante una lista de verificación de calidad, que incluyó las siguientes preguntas:

- ¿El artículo aborda tendencias en lenguajes de programación en la industria del *software*?

- ¿El artículo aborda tendencias en la gestión de bases de datos en la industria del *software*?

Cada criterio fue calificado con las opciones: “sí”, “parcialmente” o “no”, con el fin de establecer un filtro riguroso para seleccionar los documentos con mayor relevancia científica.

Extracción de datos

De los artículos seleccionados se extrajeron los siguientes datos:

- Tendencias en lenguajes de programación en la industria del *software*.
- Tendencias en sistemas de gestión de bases de datos en la industria del *software*.

La información fue sistematizada en una matriz de extracción, lo cual permitió organizar y clasificar los hallazgos de manera comparativa.

Análisis de datos

El análisis de los datos recopilados se llevó a cabo mediante un proceso de síntesis temática y categorización, en el cual se agruparon los hallazgos en torno a dos ejes principales: (i) lenguajes de programación y (ii) sistemas

de gestión de bases de datos. Posteriormente, se realizó un contraste entre las diferentes fuentes con el objetivo de identificar patrones comunes, divergencias y vacíos de conocimiento.



El procedimiento incluyó las siguientes etapas:

- **Codificación inicial:** lectura detallada de los artículos y extracción de conceptos clave asociados a tendencias tecnológicas.
- **Agrupación temática:** clasificación de los hallazgos en categorías emergentes relacionadas con popularidad, adopción, innovación y obsolescencia tecnológica.
- **Comparación transversal:** análisis de la frecuencia y recurrencia de las tendencias reportadas en las distintas

bases de datos y publicaciones.

- **Síntesis narrativa:** elaboración de un discurso integrador que articula los resultados en torno a las dinámicas actuales del sector productivo.

Este enfoque permitió no solo identificar las tendencias predominantes en lenguajes de programación y sistemas de bases de datos, sino también analizar su impacto en el desarrollo de *software*, lo que ofrece un panorama integral y actualizado de la evolución tecnológica en la industria.

RESULTADOS

A continuación, se sintetizan los principales aportes de la literatura académica sobre este tema.

Evolución y popularidad de lenguajes de programación

El análisis de repositorios y foros como GitHub y Stack Overflow muestra que Python, JavaScript y Java dominan en popularidad y uso, con crecimiento sostenido en los últimos años (Tambad et al., 2020; Dvořák y Pergl 2022). Además, la revisión estadística realizada por Orłowska et al. (2021) sobre el uso de lenguajes en proyectos de *software* libre respalda esta tendencia, al hacer evidente que Python, JavaScript y Java no solo mantienen una alta presencia en comunidades como GitHub, sino que también presentan una adopción estable en el

tiempo. Esta evidencia reafirma que la persistencia de ciertos lenguajes en el entorno del *software* libre tiene implicaciones directas en su demanda profesional y, por tanto, en su pertinencia dentro de los programas de formación en desarrollo de *software*.

En esta misma sintonía, el *ranking* TIOBE, basado en búsquedas y métricas *online*, confirma la consolidación de Python como lenguaje más popular desde el 2021, desplazando a Java y C en rentabilidad y popularidad en la comunidad (Cass, 2021).



Lenguajes de programación más utilizados

El estudio de Layedra et al. (2022) destaca que lenguajes como Python, Java, JavaScript, C# y Swift concentran la mayor demanda en el ámbito del desarrollo de aplicaciones web y móviles. Asimismo, Wang et al. (2025) afirman que JavaScript es el lenguaje de programación más popular según un informe de GitHub del 2023. Con base en los índices TIOBE y PYPL, se argumenta que estos len-

guajes poseen un equilibrio entre facilidad de aprendizaje, robustez y aplicabilidad en distintos sectores, desde el desarrollo empresarial hasta la ciencia de datos.

Los lenguajes C y C++, aunque más complejos y cercanos al *hardware*, continúan siendo relevantes para el desarrollo y se sugieren como parte de rutas de especialización o cursos avanzados (Layedra et al., 2022).

Lenguajes emergentes

Lenguajes como Go, Rust y Swift han experimentado un crecimiento significativo en comunidades técnicas como Stack Overflow y GitHub en los últimos años, lo que refleja un interés cada vez mayor por paradigmas de programación modernos, especialmente aquellos orientados a la concurrencia, la seguridad en tiempo de compilación y

la eficiencia en sistemas embebidos y distribuidos (Chakraborty et al., 2021). Este fenómeno responde a necesidades concretas de la industria actual, que demanda lenguajes capaces de ofrecer alto rendimiento sin comprometer la seguridad ni la mantenibilidad del código.

Python: versatilidad y accesibilidad

Python se ha consolidado como uno de los lenguajes más populares gracias a su sintaxis clara, su enfoque multiparadigma y su extenso ecosistema de librerías. Tal como exponen Arias et al. (2024), Python se utiliza en áreas como desarrollo web, automatización, inteligencia artificial y ciencia de datos. Además, su portabilidad y comunidad activa lo convierten en una excelente herramienta tan-

to para principiantes como para profesionales.

Sin embargo, se advierten limitaciones como su menor velocidad de ejecución frente a lenguajes compilados como Java o C++, y un alto uso de memoria (Arias et al., 2024), lo cual es relevante al elegir herramientas para proyectos con altos requerimientos de rendimiento.

Desarrollo de aplicaciones móviles: lenguajes y metodologías

En el desarrollo de aplicaciones móviles, las decisiones tecnológicas están condicionadas

por las metodologías ágiles y la necesidad de productividad. De acuerdo con Aliaga et al.





(2024), el lenguaje Kotlin se posiciona como una alternativa a Java en el ecosistema Android, al ofrecer una sintaxis más concisa y moderna.

El estudio también evidencia una alta congruencia entre lenguajes modernos y

principios de diseño orientados a la reutilización y la seguridad, elementos clave para el desarrollo sostenible de *software* móvil (Aliaga et al., 2024).

Frameworks y librerías de desarrollo

Los *frameworks* y librerías de desarrollo desempeñan un papel central en la construcción de *software*, ya que permiten acelerar el desarrollo, estandarizar arquitecturas y facilitar la escalabilidad de aplicaciones. Estas herramientas proporcionan componentes reutilizables, abstracciones funcionales y estructuras de proyecto que optimizan el trabajo de los desarrolladores, reduciendo significativamente el tiempo de desarrollo y los errores recurrentes.

En el ámbito del desarrollo web, *frameworks* como React, Angular y Vue.js se destacan por ofrecer arquitecturas de *front-end* reactivas y modulares, fundamentales para la creación de interfaces de usuario dinámicas y adaptables. React, desarrollado por Facebook, ha ganado una comunidad global robusta gracias a su enfoque declarativo y su rendimiento eficiente mediante el DOM virtual. Angular, por su parte, ofrece un ecosistema más estructurado y completo, lo que lo hace ideal para proyectos empresariales de gran escala. Vue.js, aunque más reciente, ha ganado popularidad por su curva de aprendizaje amigable y su flexibilidad para integrarse en proyectos existentes.

En el desarrollo *backend*, *frameworks* como Node.js (basado en JavaScript) y Spring (basado en Java) son ampliamente utilizados. Node.js destaca por su arquitectura orientada a eventos y su capacidad de manejar múltiples conexiones simultáneamente con un solo hilo, lo que lo hace adecuado para aplicaciones en tiempo real. Spring Framework, por su parte, es una plataforma madura y estable que facilita el desarrollo de aplicaciones Java empresariales mediante la inyección de dependencias, la programación orientada a aspectos y su integración con múltiples tecnologías.

La popularidad y el uso de estos *frameworks* están estrechamente ligados a la demanda laboral. Diversos estudios, como el de Tambad et al. (2020), han demostrado que existe una correlación directa entre la actividad en plataformas como GitHub y Stack Overflow y la preferencia por ciertos entornos de desarrollo. En particular, los autores analizaron características comunitarias y técnicas que muestran cómo los *frameworks* con mayor actividad en estas plataformas suelen coincidir con aquellos más solicitados en vacantes laborales, revelando así su alta relevancia industrial.



Comparación técnica: Python vs. Java

El análisis de Cuéllar-Juárez y Mamani-Vilca (2023) muestra una comparación técnica entre Python y Java en el contexto del desarrollo web. Python sobresale por su rapidez en la implementación y curva de aprendizaje, mientras que Java ofrece ventajas en la gestión de hilos, eficiencia de ejecución y robustez de aplicaciones empresariales.

En función de estas diferencias, se sugiere seleccionar el lenguaje según las características del proyecto, el equipo de trabajo y los requerimientos de escalabilidad (Cuéllar-Juárez y Mamani-Vilca, 2023).

Tola (2023) y Đurdev (2024) realizan un análisis comparativo centrado exclusivamente en tres lenguajes de programación web modernos. Basándose en el uso histórico y las tendencias actuales en la industria del desarrollo web, las investigaciones seleccionan Python, JavaScript y PHP como los lenguajes por evaluar. En ellas se exploran y comparan las diferentes funciones de compresión de archivos e imágenes que estos tres lenguajes ofrecen, ya sea de forma nativa o mediante bibliotecas externas.

Tabla 1. Comparativo entre Python, Javascript y PHP

Característica	Python	JavaScript	PHP
Popularidad y tendencias	Es el tercer lenguaje más popular en GitHub (2023). Superó a Java en popularidad general en el 2018. Su número de preguntas en Stack Overflow ha crecido constantemente, superando a Java desde el 2017.	Es el lenguaje más popular en GitHub (2023). Se considera el lenguaje central para el desarrollo web. Su ecosistema de librerías crece tan rápidamente que ha generado el término “fatiga de JavaScript” entre los desarrolladores.	Sigue siendo uno de los lenguajes de programación web más populares y ampliamente utilizados. Fue superado en popularidad por Java en el 2016.
Complejidad del código	Se considera el menos complejo en un estudio comparativo (frente a C, C++, Java y PHP). Requiere el menor esfuerzo y tiempo de programación, y su código tiende a ser más corto y a presentar menos errores para una misma tarea.	No se analiza con métricas de Halstead en los artículos, pero se menciona que TypeScript (un superconjunto) ofrece mejor calidad y comprensibilidad del código en comparación.	Es menos complejo que C, C++ y Java, pero más complejo que Python, según un análisis con métricas de Halstead. Su dificultad de aprendizaje se considera menor que la de C++ y Java.





Característica	Python	JavaScript	PHP
Casos de uso principales	Muy popular para aprendizaje automático (<i>machine learning</i>), <i>big data</i> y análisis de datos gracias a sus librerías. También es ampliamente usado para el desarrollo web del lado del servidor y en diversas disciplinas de ingeniería.	Esencial para el desarrollo web tanto del lado del cliente (<i>frontend</i>) como del lado del servidor (con Node.js). Es un lenguaje versátil con un ecosistema robusto.	Se utiliza principalmente como un lenguaje de <i>scripting</i> del lado del servidor para el desarrollo de aplicaciones web dinámicas. La popular plataforma de e-learning Moodle está desarrollada en PHP.
Ecosistema y <i>frameworks</i>	Ecosistema robusto con librerías como Pandas y NumPy para datos, Django para web y Matplotlib para gráficos. Entornos de desarrollo comunes incluyen Jupyter y PyCharm.	El entorno de servidor más popular es Node.js. Posee una vasta cantidad de librerías y <i>frameworks</i> que evolucionan y cambian rápidamente.	Se destaca por su facilidad de uso e interoperabilidad con múltiples bases de datos, aunque no se mencionan <i>frameworks</i> específicos en los documentos.
Funciones de compresión	Ofrece módulos nativos como zlib, gzip, bz2 y lzma para texto. Para imágenes, depende de librerías externas como pypng e imageio.	No tiene funciones de compresión nativas robustas, por lo que depende de librerías externas como pako.js para texto (Deflate, Gzip) y image-compressor.js para imágenes.	Proporciona funciones nativas integradas para la compresión de texto (gzencode, bzcompress) e imágenes (imagejpeg, imagewebp, imagepng).

Fuente: elaboración propia.

Modelado y gestión de bases de datos

La revisión sobre modelado en bases de datos NoSQL detalla la predominancia del modelo orientado a documentos, seguido de los modelos orientados a columnas y a grafos, destacando su flexibilidad frente a los esquemas relacionales (Aguilar Vera et al., 2023).

Además, Khan, Ahmad et al. (2023) realizan una comparación exhaustiva entre

arquitecturas SQL y NoSQL, y concluyen que las bases de datos relacionales son adecuadas para OLTP, mientras que NoSQL destaca por su escalabilidad en sistemas de *big data* (Khan, Kumar et al., 2023). Mester (2025) presenta una evolución detallada de los sistemas de bases de datos, comenzando con los modelos relacionales (RDBMS)



y llegando hasta las soluciones NoSQL. En el ámbito de las bases de datos relacionales, se destaca el papel de gigantes tecnológicos como Oracle, IBM (con productos como DB2) y Microsoft (SQL Server). Además, se subraya la importancia de sistemas de código abierto como MySQL y PostgreSQL para facilitar el desarrollo rápido y rentable de aplicaciones web. Para abordar las necesidades del *big data*, se describe la aparición de las bases de datos NoSQL. Entre los ejemplos más relevantes se encuentran MongoDB, Cassandra, CouchDB, así como los influyentes Bigtable de Google y Dynamo de Amazon.

Subramanian y Saravanan (2024) discuten las tendencias en bases NoSQL, destacando características de *document-store*, *key-value*, columnas y grafos, y los *trade-offs* inherentes al teorema CAP.

Basándose en el estudio empírico de Paiva et al. (2025), que analizó 362 proyectos Java de código abierto populares, MySQL emerge como el sistema de gestión de bases de datos más utilizado, presente en el 55,9 % de los proyectos que emplean SGBD, seguido por PostgreSQL (46,0 %) y H2 (44,6 %). El análisis histórico de los proyectos confirma esta tendencia, con MySQL apareciendo en 137 proyectos, H2 en 114 proyectos y PostgreSQL en 102 proyectos a lo largo de su evolución. En cuanto a las bases de datos no relacionales, Redis se posiciona como la más popular (39,6 % de los proyectos) y MongoDB ocupa el segundo lugar (22,8 %), lo que demuestra la creciente adopción de tecnologías NoSQL en proyectos Java empresariales. Adicionalmente, el estudio revela que el 51,0 % de los proyectos adoptan tanto modelos relacionales como no relacionales, lo que pone en evidencia una tendencia hacia la persistencia polígota.

Frameworks, UI/UX y diseño adaptativo

El diseño UI/UX moderno ha promovido el uso de *frameworks* adaptativos que permiten desarrollar interfaces minimalistas, responsivas y escalables. Según De La Cruz y Morejón (2023), *frameworks* como React, Vue y Bootstrap permiten construir componentes reutilizables y coherentes con el enfoque *mobile first*.

También se destaca la adopción del patrón *microfrontend*, que fomenta el desarrollo distribuido e independiente de

módulos *front-end*, alineado con las necesidades de escalado horizontal y despliegue continuo (De La Cruz y Morejón, 2023).

JavaScript, por su integración con HTML5 y CSS3, continúa siendo el lenguaje predominante para estas tecnologías, reafirmando su papel central en el desarrollo web moderno.



Calidad y seguridad

Estudios que aplican análisis bayesianos revelan que los defectos de *software* no están directamente correlacionados con el lenguaje específico, sino con prácticas de ingeniería como pruebas, revisión de código y seguridad (Furia et al., 2022). La seguridad en lenguajes emergentes también ha sido tematizada por Croft et al. (2021) mediante minería en GitHub y Stack Overflow, resaltando retos específicos asociados a cada lenguaje.

Pandya y Tiwari (2023) se centran de manera exclusiva en el análisis comparativo de

los lenguajes de programación Java y Python. Se investigan las similitudes en los desafíos que enfrentan los desarrolladores con estos dos lenguajes, utilizando un conjunto de datos de 18 892 preguntas de Stack Overflow y 42 674 problemas de 22 repositorios de GitHub. Aunque el estudio menciona otros lenguajes como C#, PHP, JavaScript y R, estas menciones se realizan únicamente en el contexto de la revisión de literatura para citar estudios previos.

Integración curricular y formación

El estudio de Tambad et al. (2020) y la evidencia empírica de su presencia en comunidades sugieren que una formación académica eficaz debe integrar lenguajes populares con prácticas profesionales reales y herramientas modernas.

Asimismo, investigaciones recientes en ingeniería continua resaltan la necesidad de articular diseño y operaciones (DevOps) en el currículo para garantizar calidad, rendimiento y alineación con flujos productivos reales (Eramo et al., 2023).

Schibelius et al. (2022) analizaron cerca de 130 000 ofertas de empleo en Estados Unidos para las principales disciplinas de ingeniería (mecánica, eléctrica, civil, química y biomédica) e identificaron un conjunto claro de lenguajes de programación predominantes. Los resultados indican que C++, MATLAB y Python son los lenguajes de programación más solicitados en estas áreas. Junto

con C, estos conforman los cuatro lenguajes principales en la mayoría de las disciplinas analizadas.

Según el análisis sistemático de la literatura realizado por Tripathi (2025), que examinó 32 estudios primarios sobre educación en bases de datos NoSQL, MongoDB emerge como el sistema de gestión de bases de datos NoSQL más ampliamente utilizado en entornos educativos de nivel superior. El estudio revela que las bases de datos como MongoDB son especialmente populares debido a su flexibilidad para manejar datos semiestructurados, seguidas por las bases de datos de grafos como Neo4j, que se enseñan por su capacidad de modelar y consultar datos complejos e interconectados. Otras tecnologías significativas incluyen Cassandra para bases de datos columnares, Redis para sistemas clave-valor, y CouchDB como alternativa en bases de datos documenta-



les. Los hallazgos indican que los cursos de NoSQL frecuentemente adoptan enfoques híbridos que integran tanto sistemas NoSQL como SQL tradicionales, siendo MySQL y PostgreSQL los gestores relacionales más comúnmente combinados con tecnologías NoSQL en contextos educativos.

En cuanto a los lenguajes de programación asociados con la educación NoSQL, el estudio identifica que los lenguajes de consulta específicos de NoSQL dominan el panorama educativo, incluyendo el lenguaje de consulta de MongoDB, Cassandra Query Language (CQL), y Cypher para Neo4j, los cuales permiten a los estudiantes realizar operaciones CRUD en diversos sistemas

NoSQL mientras trabajan con formatos de datos como JSON y XML. El análisis revela que Python se utiliza frecuentemente en conjunción con tecnologías NoSQL, particularmente en contextos de *big data* y computación en la nube, mientras que JavaScript aparece como lenguaje relevante debido a la naturaleza nativa de JSON de muchas bases de datos documentales como MongoDB. Adicionalmente, se observa una tendencia hacia la integración de *frameworks* de *big data* como Hadoop MapReduce y Apache Spark en los currículos NoSQL, lo que refleja la convergencia entre tecnologías NoSQL y ecosistemas de procesamiento de datos a gran escala.

Tendencias en el mercado laboral en Colombia

Para llevar a cabo el análisis de tendencias del mercado laboral, se implementó un *script* de *web scraping*. Este *script* se utilizó para la recolección de datos de plataformas de empleo en línea, con el que se analizaron

las descripciones de más de 613 ofertas de trabajo para desarrolladores de *software* en Colombia durante el primer semestre del 2025. Los pasos aplicados en la recopilación de datos fueron los siguientes:

Tabla 2. Pasos realizados para recopilación de ofertas laborales

Acción	Descripción
Revisión de ofertas de empleo	Bolsas de empleo: El Empleo.com, Computrabajo.com
Web <i>scraping</i>	Automatización de la consulta mediante Python
Obtención de datos	613 ofertas de empleo analizadas
Procesamiento	ETL (Extracción, Transformación, Carga). Se obtuvieron 530 registros filtrados.
Generación de gráficos y nube de palabras	Se destacan los lenguajes y <i>frameworks</i> .

Fuente: elaboración propia.

Nube de palabras

En la nube de palabras, se puede observar que los lenguajes de programación mencio-

JavaScript es el lenguaje con mayor representación, consolidándose como la habilidad más solicitada. Esto se relaciona con su papel central en el desarrollo *frontend* y *backend* (Node.js), así como en *frameworks* como React y Angular.

Java mantiene una fuerte presencia, lo que pone en evidencia su relevancia en entornos empresariales, sistemas distribuidos y aplicaciones móviles (Android).

Otros lenguajes visibles incluyen C#, TypeScript, PHP, Kotlin, Swift y C++, lo que indica la necesidad de perfiles versátiles que dominen múltiples entornos.

35



Frameworks y librerías tecnológicas

Los *frameworks* más destacados son React, Angular y Spring, asociados al desarrollo *frontend* y *backend* respectivamente.

Node.js, Vue, NestJS, Laravel y Django también tienen una presencia destacada, lo que refleja su uso en arquitecturas modernas y microservicios.

Bases de datos

La combinación de SQL y NoSQL se pone en evidencia como una competencia clave. Palabras como MySQL, PostgreSQL, MongoDB, Oracle, Cassandra y GraphQL resaltan la necesidad de manejar múltiples paradigmas

de persistencia de datos. NoSQL aparece destacado, lo que sugiere la creciente importancia de bases de datos no relacionales en arquitecturas escalables y de *big data*.

Herramientas de DevOps e infraestructura

Se mencionan en las ofertas laborales herramientas como Docker, Jenkins, Kubernetes, Azure, AWS y Terraform, lo cual señala la integración de prácticas DevOps en los

perfiles solicitados. También figuran herramientas de gestión y control de versiones como Git y GitHub, así como plataformas como Visual Studio.

Estándares web y lenguajes de marcado

Tecnologías como HTML, HTML5, CSS, CSS3, SCSS, Bootstrap y Tailwind indican una fuerte orientación hacia el diseño adaptable, la adaptabilidad y la experiencia de

usuario (UI/UX). Esto también refuerza el enfoque mobile-first que predomina en los desarrollos actuales.

Otras tendencias emergentes

Se encuentran términos como Redux, RxJS, Microservicios, GraphQL, API REST, DevOps y Cloud, que evidencian la transición hacia arquitecturas modernas, desacopladas y orientadas a servicios.

Aparecen lenguajes y entornos tradicionales (ej., COBOL, ABAP, VB y JCL), lo que sugiere que aún persisten oportunidades laborales en el mantenimiento de sistemas legados.



La nube de palabras confirma una clara tendencia hacia lenguajes modernos como JavaScript, Python y Java, así como hacia *frameworks*, *frontend* y *backend* altamente especializados. Las competencias en bases de datos híbridas, herramientas DevOps y

diseño web adaptable son cruciales en los perfiles más demandados. La presencia de tecnologías heredadas también indica la necesidad de formación flexible que prepare tanto para entornos actuales como para contextos de migración tecnológica.

Categorías de tecnologías más recurrentes en ofertas laborales

De acuerdo con los términos identificados en la nube de palabras, las tecnologías más recurrentes se pueden agrupar en:

- Lenguajes: JavaScript, Python, Java, TypeScript, C#, Kotlin
- *Frameworks*: React, Angular, Node.js, Vue, Spring

- Bases de datos: SQL, NoSQL, MySQL, MongoDB, PostgreSQL
- Herramientas: GitHub, Docker, Jenkins, Jira, Visual Studio
-

Bases de datos

En cuanto a bases de datos, los lenguajes más solicitados en las ofertas laborales se presentan en la figura 2.

La gráfica presentada muestra la frecuencia de aparición de distintas tecnologías de bases de datos mencionadas en las ofertas laborales analizadas. Se observa un claro predominio del término SQL, con un total de 75 menciones, lo que evidencia la vigencia y transversalidad del lenguaje de consulta estructurado en el desarrollo de *software*. Esta alta frecuencia sugiere que, independientemente del motor de base de datos utilizado, el conocimiento de SQL sigue siendo una competencia esencial en los perfiles profesionales del área tecnológica.

Entre las tecnologías específicas, destacan MySQL (17 menciones) y Oracle (13

menciones), que lideran entre las bases de datos relacionales concretas más demandadas. MySQL es ampliamente adoptada en desarrollos de código abierto y soluciones web, mientras que Oracle se mantiene como una opción dominante en entornos empresariales de misión crítica. PostgreSQL, con 12 menciones, también aparece con una presencia relevante, lo que confirma su creciente adopción como alternativa robusta y extensible a las opciones comerciales tradicionales.

Por otro lado, la presencia de tecnologías NoSQL como MongoDB y Cassandra (ambas con 3 menciones) indica que, si bien su demanda aún es moderada, estas bases de datos están posicionándose estratégicamente en proyectos que requieren alta escalabilidad, disponibilidad y manejo flexible de

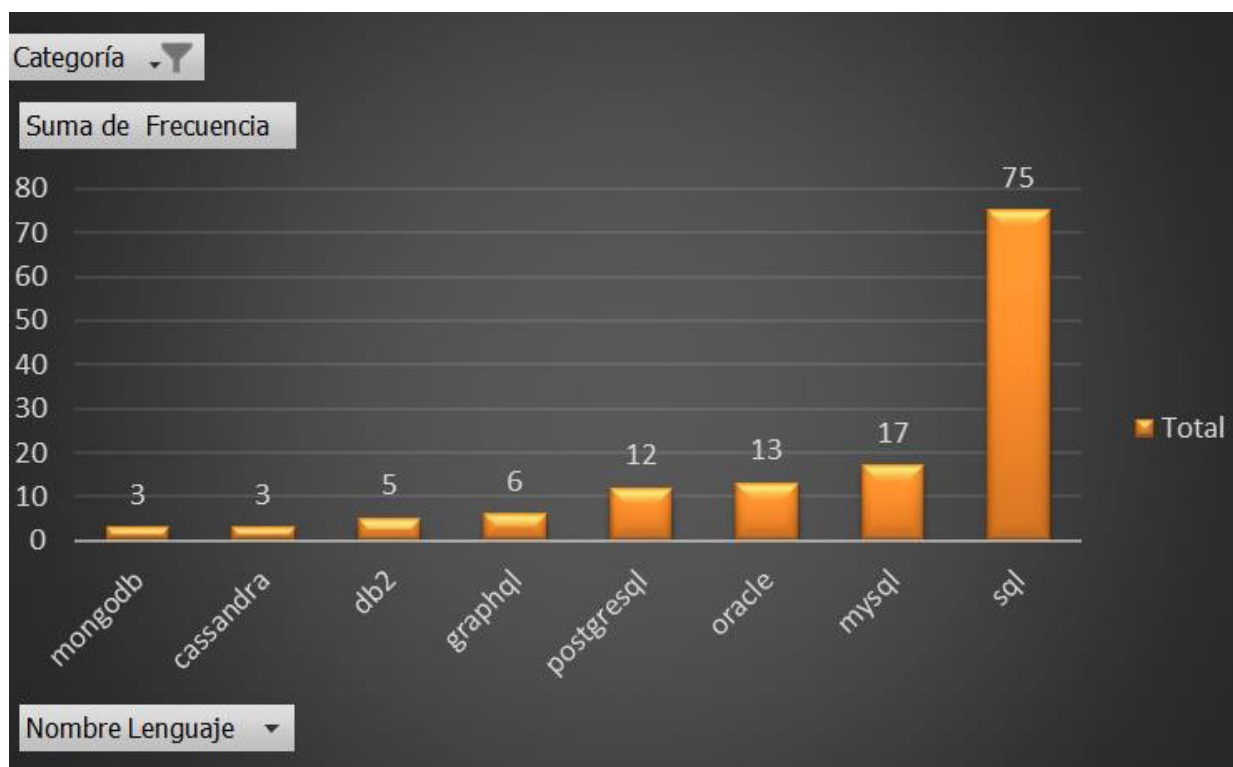


datos no estructurados. Asimismo, GraphQL aparece con 6 menciones, reflejando la adopción creciente de esta tecnología como mecanismo moderno de consulta, especialmente útil en arquitecturas desacopladas entre *frontend* y *backend*.

Finalmente, la aparición de DB2 (5 menciones) sugiere que todavía existen requerimientos de mantenimiento y operación de

sistemas legados, particularmente en sectores como la banca y los grandes sistemas transaccionales. En conjunto, estos resultados muestran que, aunque las bases de datos relacionales dominan el panorama, las tecnologías emergentes también comienzan a tener un lugar en las competencias exigidas por el mercado laboral.

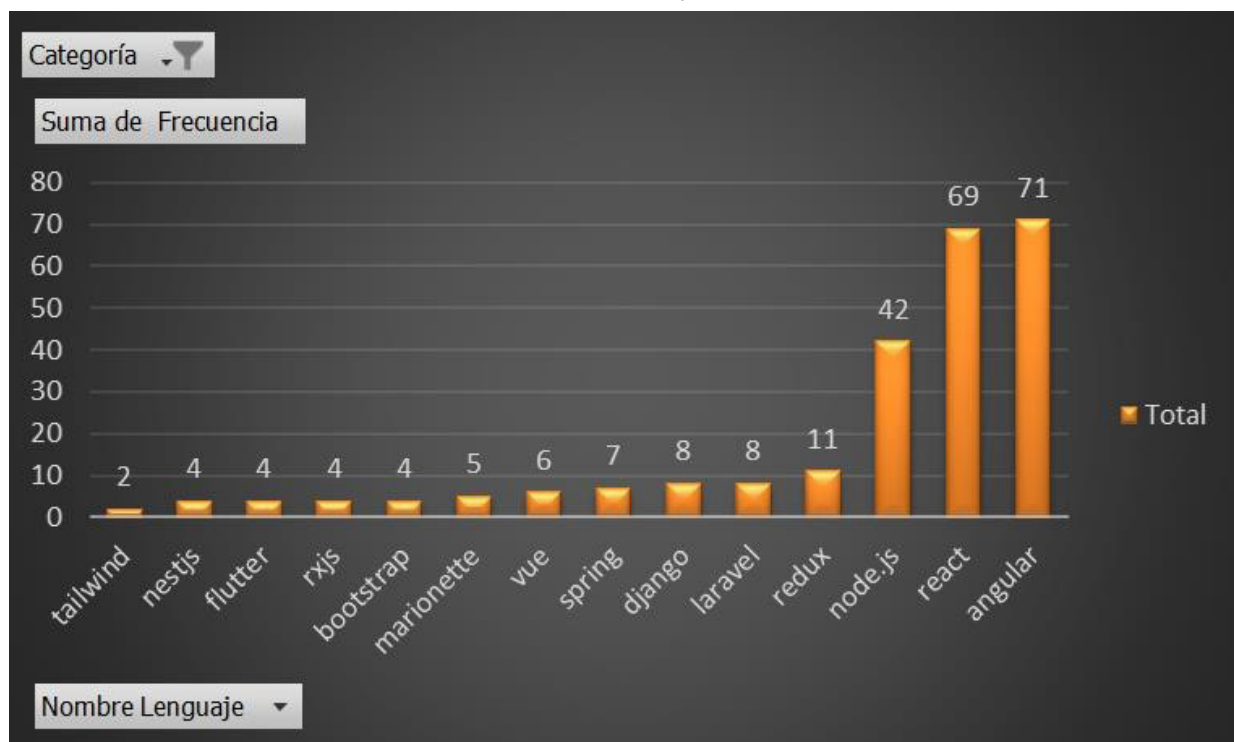
Figura 2. Ofertas laborales en bases de datos



Fuente: elaboración propia.

Frameworks y librerías

En cuanto a *frameworks* y librerías específicas solicitadas, en la figura siguiente se muestran las más solicitadas:

Figura 3. *Frameworks* y librerías

Fuente: elaboración propia.

La figura 3 muestra la frecuencia con la que distintos *frameworks*, librerías y entornos de ejecución aparecen mencionados en descripciones de vacantes para desarrolladores. Los resultados reflejan las preferencias del mercado y permiten identificar las herramientas más valoradas en el desarrollo *frontend*, *backend* y de aplicaciones móviles.

Los *frameworks* y bibliotecas más destacados son Angular (71 menciones) y React (69 menciones), lo que confirma su liderazgo en el desarrollo *frontend* moderno. Estas tecnologías permiten crear interfaces web dinámicas, escalables y responsivas, lo que las convierte en habilidades esenciales para desarrolladores de interfaces de usuario (UI). La escasa diferencia entre ambas sugiere que los empleadores valoran la

experiencia en cualquiera de las dos, aunque algunas vacantes podrían requerir conocimiento específico según el stack tecnológico de la empresa.

Node.js, con 42 menciones, ocupa el tercer lugar. Su presencia indica la demanda de perfiles *full-stack* o *backend* capaces de utilizar JavaScript en el servidor, lo que favorece arquitecturas isomórficas y asíncronas. Este entorno ha cobrado fuerza gracias a su ecosistema de paquetes, su eficiencia para manejar múltiples solicitudes simultáneas y su capacidad de integración con herramientas modernas.

Redux (11 menciones), que actúa como biblioteca de gestión de estado comúnmente utilizada junto con React, y Laravel y Django



(8 menciones cada uno), orientados al desarrollo *backend* con PHP y Python respectivamente, también ponen en evidencia una demanda intermedia. Esto indica la existencia de vacantes más especializadas que requieren conocimientos más allá del *frontend*.

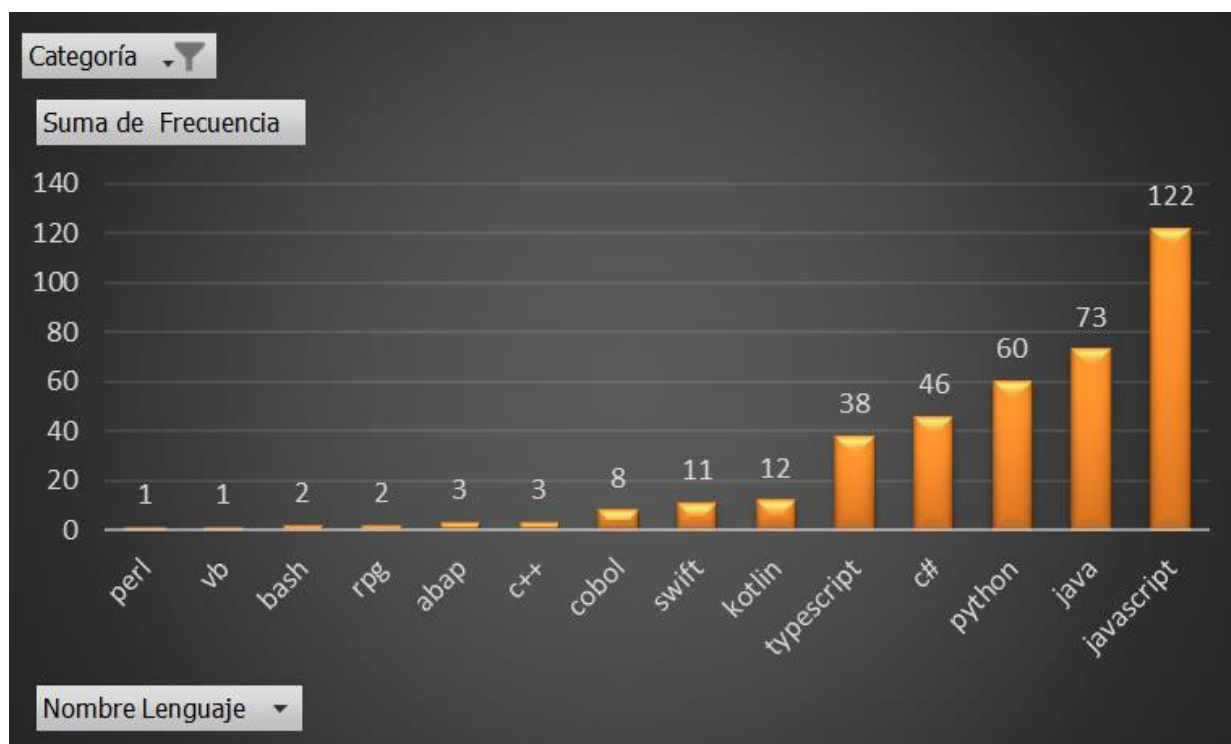
Otros *frameworks* y bibliotecas como Spring (7 menciones), Vue (6 menciones), Marionette (5 menciones), Bootstrap, RxJS, Flutter y NestJS (4 menciones cada uno),

aunque con menor frecuencia, demuestran una tendencia a la diversificación tecnológica, en la que los desarrolladores deben adaptarse a múltiples herramientas según las necesidades del proyecto. Finalmente, Tailwind (2 menciones) aparece con menor relevancia, aunque su adopción puede estar creciendo entre equipos que priorizan un control total sobre el diseño sin depender de componentes preconstruidos.

Lenguajes de programación

En cuanto a lenguajes de programación, en la figura 4 se muestran los más solicitados:

Figura 4. Ofertas por lenguajes de programación



Fuente: elaboración propia.



La gráfica muestra la frecuencia con la que distintos lenguajes de programación fueron mencionados en descripciones de vacantes laborales relacionadas con el desarrollo de *software*. Este análisis permite identificar cuáles son las habilidades técnicas más valoradas actualmente en el mercado.

El lenguaje con mayor número de menciones es JavaScript, con un total de 122 apariciones, lo que lo consolida como el lenguaje más demandado. Su versatilidad —tanto en desarrollo *frontend* como *backend*, mediante *frameworks* y bibliotecas como React, Angular y Node.js— lo convierte en una herramienta esencial para desarrolladores web de todos los niveles.

En segundo lugar, se encuentra Java (73 menciones), lo que reafirma su papel fundamental en aplicaciones empresariales, sistemas Android y arquitecturas escalables. Su madurez y robustez hacen que siga siendo ampliamente adoptado en sectores financieros, gubernamentales y educativos.

Python, con 60 menciones, ocupa el tercer lugar. Este lenguaje ha incrementado su popularidad gracias a su sintaxis simple, su ecosistema científico (por ejemplo, en ciencia de datos, inteligencia artificial y aprendizaje automático), y sus *frameworks backend* como Django y Flask. Su inclusión en proyectos académicos y de prototipado rápido también lo posiciona como una tecnología prioritaria en entornos de innovación.

El lenguaje C# (46 menciones) muestra una alta frecuencia relativa, especialmente en desarrollos vinculados al ecosistema Microsoft (.NET, Azure, aplicaciones de escritorio y videojuegos con Unity). Su presencia demuestra que aún es un pilar tecnológico en organizaciones que trabajan con tecnologías corporativas integradas.

Lenguajes como TypeScript (38 menciones) reflejan una evolución de JavaScript, ya que ofrecen tipado estático y mayor seguridad en proyectos a gran escala. Su adopción en aplicaciones modernas demuestra una tendencia hacia mejores prácticas de desarrollo y mantenimiento de código.

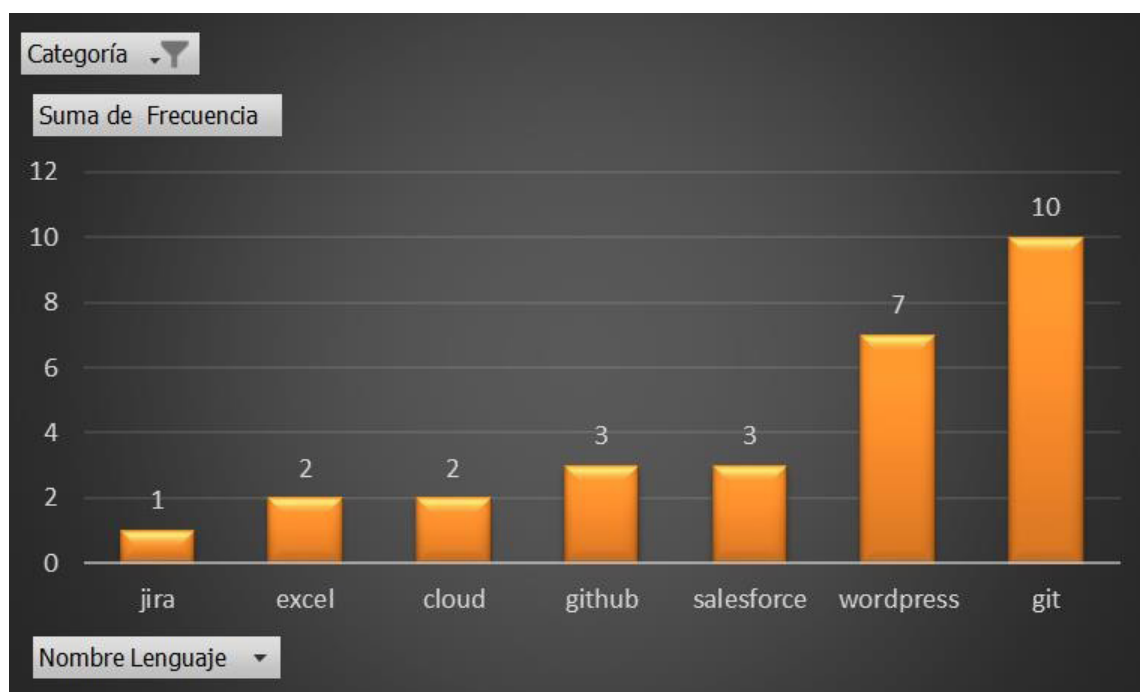
Más abajo en la gráfica aparecen Kotlin (12 menciones) y Swift (11 menciones), lo cual indica la demanda de perfiles especializados en desarrollo móvil nativo para Android e iOS, respectivamente. Esto refuerza la necesidad de formación en tecnologías móviles además de las tecnologías web.

Finalmente, lenguajes como COBOL, C++, ABAP, RPG, Bash, VB y Perl tienen una frecuencia baja (entre 1 y 8 menciones), lo que indica su uso en sistemas heredados o nichos específicos, como el mantenimiento de sistemas financieros antiguos o en entornos *mainframe*. Aunque tienen una menor demanda general, su conocimiento puede representar una ventaja estratégica en sectores específicos donde aún se mantienen en producción.

Herramientas y plataformas

En cuanto al uso de plataformas, se presentan las siguientes:



Figura 5. Ofertas en herramientas y plataformas

Fuente: elaboración propia.

La figura 5 refleja la frecuencia de aparición de distintas herramientas y plataformas de apoyo utilizadas en el ciclo de desarrollo de *software*, tal como se identificaron en las descripciones de ofertas laborales analizadas.

En primer lugar, Git destaca con 10 menciones, consolidándose como la herramienta de control de versiones más requerida. Su capacidad para gestionar el historial de cambios, facilitar el trabajo colaborativo entre equipos distribuidos y permitir su integración con plataformas como GitHub o GitLab lo convierte en una competencia fundamental para cualquier desarrollador. Su presencia en los procesos de integración y entrega continua (CI/CD) refuerza su carácter transversal en distintos entornos de desarrollo.

WordPress aparece en segundo lugar, con 7 menciones, lo que sugiere una demanda significativa de conocimientos sobre gestión de contenidos (CMS) para el desarrollo de sitios web y soluciones empresariales basadas en plantillas. Su inclusión en la muestra indica que aún existen oportunidades laborales para perfiles con enfoque web orientado al diseño, la implementación rápida y la personalización de plataformas.

Salesforce y GitHub, con 3 menciones cada uno, reflejan la necesidad de dominar plataformas de gestión de versiones y entornos de desarrollo colaborativo. GitHub, además de ser un repositorio ampliamente utilizado, permite la integración con flujos DevOps y la automatización de despliegues. Por su parte, Salesforce representa el conocimiento en CRM (Customer Relationship Manage-



ment), enfocado en soluciones empresariales de ventas, atención al cliente y *marketing*. Con 2 menciones cada una, Cloud y Excel muestran un nivel moderado de demanda. La mención a “Cloud” probablemente alude a competencias básicas en computación en la nube (como el uso de servicios en AWS, Azure o Google Cloud), mientras que Excel mantiene su relevancia como herramienta complementaria para tareas de análisis, prototipado rápido o integración con otras soluciones.

Finalmente, Jira, con una sola mención, aparece como una herramienta de gestión de proyectos y seguimiento de tareas bajo metodologías ágiles (Scrum y Kanban). Aunque su presencia es baja en esta muestra, en contextos organizacionales ágiles Jira sigue siendo ampliamente utilizado, por lo cual su dominio puede aportar valor adicional en roles de coordinación y desarrollo colaborativo.

Tendencias mundiales en uso de bases de datos

Para complementar la búsqueda de ofertas laborales, se contrasta esta información con las tendencias en bases de datos y lenguajes de programación a nivel mundial.

En la figura 6 se presenta el *ranking* de popularidad de sistemas de gestión de bases de datos (SGBD) publicado por DB-Engines, consultado para julio del 2025. Este *ranking* se construye con base en una metodología que combina múltiples fuentes de datos, entre ellas, el volumen de búsquedas, las menciones en sitios web, la actividad en foros técnicos, la frecuencia en ofertas laborales y el número de perfiles profesionales asociados a cada tecnología. Su análisis longitudinal permite comprender no sólo el estado actual del uso de estas herramientas, sino también su evolución en el tiempo y su proyección en el mercado.

En la parte superior del *ranking* se mantienen de forma sostenida Oracle, MySQL y Microsoft SQL Server, lo que confirma su dominio histórico en entornos empresariales y de misión crítica. Estas bases de datos han

logrado mantener su liderazgo gracias a su madurez, confiabilidad y amplia comunidad de soporte, a pesar del surgimiento de nuevas tecnologías más flexibles o especializadas.

Por otro lado, se pone en evidencia un crecimiento notable de tecnologías de código abierto como PostgreSQL y MongoDB. PostgreSQL, en particular, ha escalado posiciones de forma progresiva, convirtiéndose en una opción preferida para organizaciones que requieren estándares SQL completos, extensibilidad y herramientas modernas sin incurrir en costos de licenciamiento privativo. MongoDB, en cambio, representa la consolidación del paradigma NoSQL, ideal para manejar datos no estructurados, escalar horizontalmente y adaptarse a arquitecturas ágiles y distribuidas.

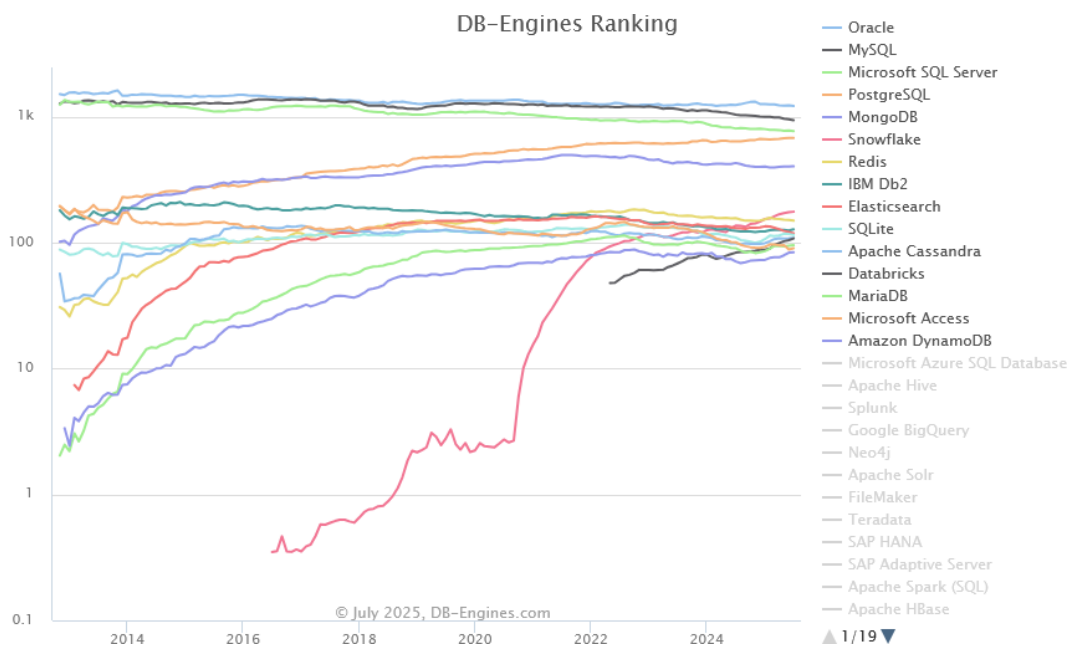
Un fenómeno especialmente llamativo es el vertiginoso ascenso de Snowflake, que desde el 2018 ha irrumpido en el grupo superior del *ranking*, consolidándose como una de las soluciones más relevantes en almacenamiento y análisis de datos en la nube.



Su arquitectura desacoplada entre almacenamiento y cómputo, así como su facilidad de integración en entornos de inteligencia de negocios, lo han posicionado como una plataforma clave en la analítica moderna. De

igual manera, herramientas como Databricks, basadas en Apache Spark, muestran una tendencia positiva, impulsadas por su uso intensivo en ciencia de datos, *machine learning* y *big data*.

Figura 6. Tendencias mundiales en bases de datos según DB-Engines



Fuente: elaboración propia.

En contraste, tecnologías tradicionales como IBM Db2, SQLite, Microsoft Access y MariaDB presentan una tendencia más estable o en ligero descenso, lo cual puede interpretarse como una señal de estancamiento en su adopción. Aunque siguen siendo ampliamente utilizadas —particularmente en sistemas heredados, entornos académicos o aplicaciones ligeras—, han cedido protagonismo frente a tecnologías más adaptadas a la nube, el tiempo real y los grandes volúmenes de datos.

Según el estudio de Nissan et al. (2025), que presenta ANOC (Automated NoSQL Car-

ver), las bases de datos NoSQL analizadas representan sistemas clave-valor ampliamente utilizados en entornos empresariales y de investigación forense. El análisis empírico se centra en cuatro sistemas de gestión de bases de datos NoSQL representativos: Berkeley DB, ZODB (Zope Object Database), etcd y LMDB (Lightning Memory-Mapped Database). Estos SGBD fueron seleccionados por su relevancia en aplicaciones de producción y su diversidad arquitectónica. Berkeley DB y LMDB son sistemas basados en páginas con directorios de registros; etcd un sistema distribuido de almacenamiento clave-valor utilizado en infraestructuras de contenedo-



res como Kubernetes, y ZODB un sistema de almacenamiento de objetos serializados utilizado principalmente en aplicaciones Python. Los experimentos fueron realizados en múltiples versiones y sistemas operativos, incluyendo Berkeley DB 18.1.40, ZODB 6.0, etcd 3.2.26 y LMDB 0.9.24 en entornos Linux y Windows.

Este escenario pone en evidencia una diversificación del ecosistema de bases de datos, en el que ya no predomina un único

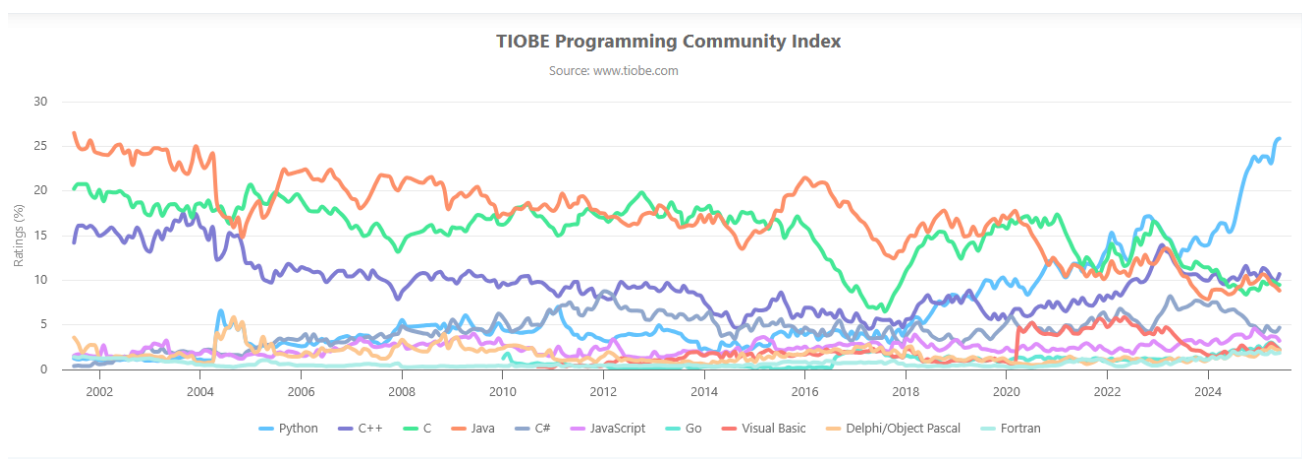
modelo o tecnología. Las organizaciones adoptan soluciones distintas según el tipo de datos que manejan, la naturaleza del proyecto, la escalabilidad requerida y la integración con herramientas modernas de análisis. Esta tendencia plantea un reto importante a los procesos formativos: se requiere una formación académica que combine fundamentos sólidos en bases relacionales clásicas con habilidades en bases NoSQL, servicios en la nube y arquitecturas orientadas a datos.

Tendencias mundiales en lenguajes de programación

En cuanto a los lenguajes de programación se tiene el dato del TIOBE Programming Community Index que presenta la evolución en la popularidad de los principales lenguajes de programación desde el 2001 hasta julio del 2025. Este índice, basado en el nú-

mero de resultados encontrados en motores de búsqueda, artículos, foros técnicos, cursos y certificaciones, es un referente global para medir la presencia activa y el interés de la comunidad en torno a cada lenguaje. En la figura 7 se pueden apreciar estas tendencias.

Figura 7. Tendencias en lenguajes de programación



Fuente: TIOBE (2026).

En la figura 7, el dato más sobresaliente de esta evolución es el ascenso de Python, que a partir del 2018 inicia un crecimiento sos-

tenido hasta posicionarse como el lenguaje más popular del mundo desde el 2021. Para mediados del 2025, Python supera el 25 % de



participación en el índice, lo que refleja su protagonismo en áreas estratégicas como la inteligencia artificial, la ciencia de datos, el desarrollo web, la automatización y el aprendizaje automático. Esta popularidad también se ha visto reforzada por su sintaxis simple, su enfoque multiparadigma y una comunidad amplia y activa, lo que lo convierte en el lenguaje de entrada más recomendado tanto en educación formal como en autoformación.

En contraste, lenguajes como Java y C, que dominaron el *ranking* durante los primeros quince años del siglo XXI, han disminuido su participación. Java, aunque sigue siendo uno de los lenguajes más utilizados a nivel empresarial, muestra una caída progresiva en los últimos años. Su uso persiste especialmente en sistemas corporativos, desarrollo de aplicaciones móviles para Android y entornos educativos con orientación a la programación orientada a objetos. Por su parte, el lenguaje C continúa vigente en ámbitos como los sistemas embebidos y el desarrollo de *software* de bajo nivel, aunque su adopción generalizada ha disminuido frente a lenguajes más modernos y versátiles.

El lenguaje C++ ha mostrado una estabilidad relativa con ligeras oscilaciones a lo largo del tiempo. Su fortaleza radica en su uso en el desarrollo de videojuegos, *software*

de alto rendimiento y aplicaciones que exigen un control estricto de recursos. C#, en cambio, ha mantenido una presencia constante gracias al ecosistema de Microsoft y a su uso en plataformas como .NET y Unity para el desarrollo de aplicaciones empresariales y videojuegos.

JavaScript, aunque es uno de los lenguajes más utilizados en el desarrollo web, aparece en el índice con una participación moderada en comparación con su uso real en la industria. Esta aparente discrepancia puede deberse a que el índice TIOBE mide la presencia en documentación formal y resultados de búsqueda, en los que lenguajes como Java o C tienen históricamente mayor volumen. Aun así, JavaScript sigue siendo esencial para el desarrollo *frontend* y ha ampliado su dominio hacia el *backend* con Node.js.

Finalmente, lenguajes como Visual Basic, Delphi/Object Pascal y Fortran presentan una clara tendencia descendente. Estos lenguajes, aunque aún están presentes en sistemas heredados o nichos específicos, han perdido relevancia en la formación y en la industria actual. Su aprendizaje solo es justificado en contextos particulares en los que su legado aún requiere soporte o mantenimiento.

Desarrollo de *software* cuántico

El análisis combinado de las respuestas de 57 profesionales (Jiménez-Navajas et al., s. f.) y de 34 estudios primarios (Khan Ahmad et al., 2023) muestra una clara predominancia de ciertos lenguajes en el desarrollo

de *software* cuántico híbrido. En el ámbito cuántico, Qiskit se posiciona como el más utilizado, seguido por OpenQASM y Cirq, lo que refleja el liderazgo de las tecnologías impulsadas por IBM y Google. En la parte



clásica, Python domina con alrededor del 50 % de adopción, debido a su rol como base de la mayoría de bibliotecas y *frameworks* cuánticos, mientras que Java, C++ y C# mantienen una presencia menor pero significativa (entre el 9 % y el 12 %).

En cuanto a las herramientas arquitectónicas, el 64 % de las soluciones identificadas son de código abierto, y un 73 % están orientadas a la simulación. Entre las más destacadas se encuentran XACC para compilación híbrida, Strawberry Fields para computación cuántica fotónica y QuNetSim para redes cuánticas. Predominan las arquitecturas monolíticas (33 %), por encima de las multicapas (27 %) y las orientadas a servicios (25 %), lo que pone en evidencia un campo en etapa temprana de desarrollo, en el que prima la simplicidad estructural frente a patrones empresariales más sofisticados.

Los hallazgos obtenidos del análisis empírico de ofertas laborales, complementados con índices reconocidos como TIOBE y DB-Engines, permiten reflexionar sobre la evolución, consolidación y proyección de las tecnologías clave en el desarrollo de *software*. Las tendencias detectadas no solo hacen evidentes las competencias técnicas más demandadas por el mercado, sino que también orientan la pertinencia y oportunidad de los contenidos formativos en programas académicos de ingeniería y tecnología.

En primer lugar, la supremacía de JavaScript, Python y Java en las ofertas laborales se ve reforzada por su presencia constante en los principales índices globales de popularidad. JavaScript, con su capacidad de operar tanto en el *frontend* como en el *backend*,

lidera las menciones en vacantes, mientras que Python ha tomado un rol protagónico por su uso transversal en inteligencia artificial, ciencia de datos y automatización. Java, por su parte, conserva una posición sólida, especialmente en entornos corporativos y desarrollo móvil Android. Estos resultados concuerdan con el índice TIOBE, en el que Python ocupa el primer lugar desde el 2021 y desplaza a lenguajes tradicionales como C y Java, lo que confirma su ascenso como lenguaje estándar en la formación de nuevos desarrolladores.

La tendencia de diversificación tecnológica también se hace evidente. Aunque los lenguajes clásicos mantienen vigencia en nichos específicos, las empresas demandan cada vez más conocimientos en lenguajes modernos y *frameworks* emergentes, como TypeScript, Kotlin y Swift, y herramientas como React, Angular, Node.js y Spring. Estos *frameworks* no solo ofrecen velocidad de desarrollo, sino también adaptabilidad a arquitecturas modernas basadas en componentes, servicios y entornos móviles. Esta realidad requiere que las mallas curriculares se actualicen continuamente para incluir estos entornos dentro de los planes formativos, más allá del conocimiento de un lenguaje base.

En el ámbito de las bases de datos, se observa una clara dualidad entre la persistencia del modelo relacional y la irrupción progresiva de tecnologías NoSQL y basadas en la nube. Aunque SQL, MySQL, PostgreSQL y Oracle siguen liderando en frecuencia de uso, bases como MongoDB, Cassandra y Snowflake ganan espacio en proyectos distribuidos, escalables y orientados a *big data*. La gráfica del *ranking* DB-Engines



confirma esta tendencia, con PostgreSQL y MongoDB en crecimiento sostenido, y con la impresionante irrupción de Snowflake como nuevo referente en almacenamiento y análisis de datos en la nube. Esta transición implica que los perfiles de desarrollo deben incorporar habilidades híbridas para gestionar tanto bases relacionales como no relacionales, con comprensión de modelos de datos, escalabilidad y rendimiento.

CONCLUSIONES

El presente estudio permitió identificar y analizar las principales tendencias en lenguajes de programación, *frameworks*, tecnologías de bases de datos y herramientas complementarias, a partir de una metodología mixta que combinó la revisión sistemática y la minería de texto de ofertas laborales mediante técnicas de *web scraping*. Esta aproximación metodológica permitió contrastar la información teórica disponible con los requerimientos reales del mercado laboral, lo que dio lugar a insumos valiosos para la toma de decisiones curriculares y de formación.

Los resultados evidencian que lenguajes como JavaScript, Python y Java siguen siendo pilares en la industria del desarrollo de *software*. JavaScript destaca por su omnipresencia en aplicaciones web; Python, por su versatilidad en automatización, ciencia de datos e inteligencia artificial, y Java, por su robustez en entornos empresariales y móviles. Esta tendencia es consistente con los datos de los índices internacionales TIOBE y PYPL, que posicionan estos lenguajes como los más populares y con un crecimiento sostenido.

Respecto a las herramientas y plataformas, tecnologías como Git, GitHub, Jira, Salesforce y entornos de colaboración como WordPress complementan el perfil técnico requerido. Su mención en las ofertas laborales, aunque menos frecuente que la de los lenguajes, indica que el desarrollador moderno no solo debe escribir código, sino también integrarse a flujos DevOps, manejar repositorios colaborativos, y participar en la gestión de tareas bajo metodologías ágiles.

En cuanto a *frameworks* y bibliotecas, tecnologías como React, Angular, Node.js y Spring lideran la demanda en perfiles profesionales. Su adopción está directamente relacionada con arquitecturas modernas, metodologías ágiles y el enfoque *mobile-first*, por lo cual su incorporación en los planes de formación resulta fundamental para mejorar la empleabilidad de los estudiantes.

Respecto al manejo de bases de datos, se observa una fuerte presencia de sistemas relacionales tradicionales como SQL, MySQL, PostgreSQL y Oracle, pero también un crecimiento notable de tecnologías NoSQL como MongoDB y Cassandra, y soluciones en la nube como Snowflake. Esta convergencia demanda que los desarrolladores adquieran competencias híbridas que les permitan adaptarse tanto a modelos clásicos como a paradigmas distribuidos y escalables.

En el caso de las herramientas de soporte al desarrollo, Git, GitHub, Jira y Salesforce complementan el perfil técnico del desarrollador moderno, destacando la importancia del trabajo colaborativo, la automatización



de flujos y la gestión eficiente de proyectos bajo metodologías ágiles. Estos elementos, si bien no son el eje central de la programación, representan habilidades esenciales en entornos profesionales contemporáneos.

Las tendencias identificadas sugieren la necesidad de investigaciones longitudinales que evalúen la evolución de la adopción tecnológica conforme el paradigma cuántico madure hacia aplicaciones comerciales. Resulta crítico desarrollar estudios específicos sobre la integración de sistemas de gestión de datos en arquitecturas cuánticas híbri-

das, así como analizar la transición desde arquitecturas monolíticas hacia patrones más complejos a medida que las aplicaciones cuánticas alcancen escalas empresariales. Adicionalmente, se requiere investigación sobre la diversificación del ecosistema de lenguajes cuánticos y su impacto en la productividad del desarrollo, particularmente en contextos en los que la dependencia actual de Python podría representar un cuello de botella para la escalabilidad y el rendimiento de sistemas cuánticos distribuidos.

REFERENCIAS

- Aguilar Vera, R., Naal Jácome, A., Díaz Mendoza, J., y Gómez Gómez, O. (2023). NoSQL Database Modeling and Management: A Systematic Literature Review. *Revista Facultad de Ingeniería*, 32(65), e16519. <https://doi.org/10.19053/01211129.v32.n65.2023.16519>
- Aliaga Arias, J. J., Atachao Vilchez, C. A. y Armada Pacheco, J. M. (2024). Similitudes y congruencias de los lenguajes de programación en el desarrollo de aplicaciones móviles. *e-Revista Multidisciplinaria del Saber*, 2, 106. <https://doi.org/10.61286/e-rms.v2i.106>
- Arias Vivanco, D., Defas Itaz, M., y Jarrín Sánchez, E. (2024). Reseña del uso del lenguaje de programación Python en el desarrollo de software. *Revista Científica Élite*, 6(2), 1-6. <https://doi.org/10.69603/itsqmet.vol6.n2.2024.90>
- Cass, S. (2021, octubre). Top programming languages 2021. *IEEE Spectrum*. <https://spectrum.ieee.org/top-programming-languages-2021>
- Chakraborty, P., Shahriyar, R., Iqbal, A., y Uddin, G. (2021). How do developers discuss and support new programming languages in technical Q&A site? An empirical study of Go, Swift, and Rust in Stack Overflow. *Information and Software Technology*, 137, 106603. <https://doi.org/10.1016/j.infsof.2021.106603>
- Croft, R., Xie, Y., Zahedi, M., Babar, M. A., y Treude, C. (2021). An empirical study of developers' discussions about security challenges of different programming languages *Empirical Software Engineering*, 27(1), 27. <https://doi.org/10.1007/s10664-021-10054-w>



- Cuéllar-Juárez, L. N., y Mamani-Vilca, E. (2023). Análisis comparativo de lenguajes de programación: Python vs. Java en el desarrollo de aplicaciones web. *Micaela Revista de Investigación - UNAMBA*, 5(1), 82-89. <https://doi.org/10.57166/micaela.v5.n1.2024.143>
- De La Cruz Rivera, L. A. y Morejón López, G. E. (2024). Tendencia en el diseño web sobre los marcos de trabajo para esquemas adaptativos: Revisión sistemática de la literatura. *Revista Social Fronteriza*, 4(3), e282. [https://doi.org/10.59814/resofro.2024.4\(3\)282](https://doi.org/10.59814/resofro.2024.4(3)282).
- Đurdev, D. (2024). Popularity of programming languages. *AIDASCO Reviews*, 2(2), 24-29. <https://doi.org/10.59783/aire.2024.59>
- Dvořák, O. y Pergl, R. (2022). Tackling rapid technology changes by applying enterprise engineering theories. *Science of Computer Programming*, 215, 102747. <https://doi.org/10.1016/j.sci-co.2021.102747>.
- Eramo, R., Tucci, M., Di Pompeo, D., Cortellessa, V., Di Marco, A., y Taibi, D. (2023). Architectural support for software performance in continuous software engineering: A systematic mapping study. *Journal of Systems and Software*, 207, 111833. <https://doi.org/10.1016/j.jss.2023.111833>
- Furia, C. A., Torkar, R., y Feldt, R. (2022). Applying bayesian analysis guidelines to empirical software engineering data: The case of programming languages and code quality. *ACM Transactions on Software Engineering and Methodology*, 31(3), 1-38. <https://doi.org/10.1145/3490953>
- GitHub. (2023). *The state of the Octoverse 2023*. <https://github.blog/2023-11-08-the-state-of-open-source-and-ai/>
- Jiménez-Navajas, L., Bühler, F., Leymann, F., Pérez-Castillo, R., Piattini, M., y Vietz, D. (2024). Quantum software development: A survey. *Quantum Information and Computation*, 24(7-8), 609-642. <https://doi.org/10.26421/QIC24.7-8-4>
- Khan, A. A., Ahmad, A., Waseem, M., Liang, P., Fahmideh, M., Mikkonen, T., y Abrahamsson, P. (2023). Software Architecture for Quantum Computing Systems: A Systematic Review. *Journal of Systems and Software*, 201, 111682. <https://doi.org/10.1016/j.jss.2023.111682>
- Khan, W., Kumar, T., Zhang, C., Raj, K., Roy, A. M., y Luo, B. (2023). SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review. *Big Data Cogn. Comput.*, 7(2), 97. <https://doi.org/10.3390/bdcc7020097>
- Layedra Larrea, N. P., Ramos Valencia, M. V., Salazar Cazco, S. A., y Baldeón Hermida, B. A. (2022). Análisis de los lenguajes de programación más utilizados en el desarrollo de aplicaciones web y móviles. *Dominio de las Ciencias*, 8(3). <https://dialnet.unirioja.es/descarga/articulo/8635309.pdf>



- Mester, U. (2025). How database development has evolved over the years in the web development area? *International Journal of Computer Trends and Technology*, 73(3),103-11. <https://doi.org/10.14445/22312803/IJCTT-V73I3P113>
- Nissan, M I., Wagner, J., y Rasin, A. (2025). ANOC: Automated NoSQL database carver. *Forensic Science International: Digital Investigation*, 53,301929. <https://doi.org/10.1016/j.fsi-di.2025.301929>
- Orlowska, A., Chrysoulas, C., Jaroucheh, Z., y Liu, X. (2021). Programming languages: A usage-based statistical analysis and visualization. En *ICISS 2021: Proceedings of the 2021 The 4th International Conference on Information Science and Systems* (pp. 143-148). Association for Computing Machinery. <https://doi.org/10.1145/3459955.3460614>
- Paiva, C. A., Maximino, R., Paiva, F., Vieira, R. A., Espanha, N., Pimentel, J. F., Wiese, I., Gerosa, M. A., Steinmacher, I., Murta, L., y Braganholo, V. (2025). Analyzing the adoption of database management systems throughout the history of open source projects. *Empirical Software Engineering*, 30(3), 71. <https://doi.org/10.1007/s10664-025-10627-z>
- Pandya, N. y Tiwari, S. (2023). Similarities in challenges faced by developers: Investigations on Stack Overflow and GitHub. En S. Agarwal, A. Roychoudhury y R. Purandare (Eds.), *ACM International Conference Proceeding Series* (pp. 531-535). Association for Computing Machinery.
- Schibeliuss, L., Ross, A. y Katz, A. (2022, 23 de agosto). *An empirical study of programming languages specified in engineering job postings* [Ponencia]. 2022 ASEE Annual Conference & Exposition, Mineápolis, Minnesota, Estados Unidos.
- Subramanian, S., y Saravanan, S. (2024). Modern trends in no SQL data bases. *International Journal of Computer Trends and Technology*, 72(9), 126-130. <https://doi.org/10.14445/22312803/IJCTT-V72I9P119>
- Tambad, S., Nandwani, R. N., y McIntosh, S. K. (2020). *Analyzing programming languages by community characteristics on GitHub and StackOverflow* (arXiv:2006.01351). arXiv. <https://doi.org/10.48550/arXiv.2006.01351>
- TIOBE Software. (2026). *TIOBE Index for September 2026* [Gráfico]. TIOBE. <https://www.tiobe.com/tiobe-index/>
- Tola, K. L. (2023). Comparative study of compression functions in modern web programming languages. En *2023 IEEE 14th International Conference on Electromechanical and Energy Systems (SIELMEN)* (pp. 1-5). IEEE. <https://doi.org/10.1109/SIELMEN59038.2023.10290843>
- Tripathi, N. (2025). NoSQL database education: A review of models, tools and teaching methods. *Journal of Systems and Software*, 226, 112391. <https://doi.org/10.1016/j.jss.2025.112391>
- Wang, Z., Fang, Y., y Wang, N. (2025). An empirical study on bugs in TypeScript programming language. *Journal of Systems and Software*, 226, 112445. <https://doi.org/10.1016/j.jss.2025.112445>

