

Reseña del uso del lenguaje de programación Python en el Desarrollo de Software

Review of the use of the Python programming language in Software Development.

D. Arias Vivanco¹ , M. Defas Itaz¹  y E. Jarrín Sánchez¹ 

1 Instituto Tecnológico Superior Quito Metropolitano. Carán N3-195 y Calle B (Nueva Tola 2) Quito, Ecuador.;

daarias@itsmqmet.edu.ec, madefas@itsqmet.edu.ec, ejarrins@itsmqmet.edu.ec

Resumen: Introducción: En este artículo se define de manera concisa y clara el lenguaje de programación Python, resaltando sus características principales y su importancia en el panorama actual del desarrollo de software. Python, conocido por su sintaxis sencilla y legible, ha ganado popularidad debido a su versatilidad y facilidad de uso.

Objetivo: El objetivo principal de este estudio es responder a la pregunta: ¿Qué papel juega Python en el desarrollo de software actual y cuáles son sus perspectivas futuras? Este análisis pretende comprender el impacto de Python en la industria del software y evaluar cómo su evolución puede influir en futuras tendencias tecnológicas.

Métodos: Para abordar este objetivo, se emplearon métodos de revisión bibliográfica para recopilar información relevante sobre Python. Se establecieron criterios de selección rigurosos para asegurar la calidad y actualidad de las fuentes consultadas, y se realizó un análisis de datos que combinó enfoques cualitativos y cuantitativos.

Resultados: El artículo resume las principales ventajas y desventajas de Python, además de explorar sus diversas áreas de aplicación. Python ofrece ventajas como facilidad de aprendizaje y un ecosistema robusto de bibliotecas, pero también presenta desventajas como menor velocidad de ejecución en comparación con otros lenguajes.

Conclusiones: En conclusión, el estudio sintetiza los resultados obtenidos, evalúa el uso actual de Python en el desarrollo de software y ofrece recomendaciones prácticas para los desarrolladores, destacando la importancia de mantenerse actualizado con las últimas versiones y herramientas disponibles.

Palabras clave: Python, Lenguaje de programación, Evolución tecnológica, Sintaxis sencilla, Desarrollo de software, herramientas de desarrollo.

ÉLITE 2024, VOL. (6). NÚM. (2)
ISSN: 2600-5875

Recibido: 09/09/2024
Revisado: 15/09/2024
Aceptado: 18/09/2024
Publicado: 29/09/2024

Abstract: Introduction: This article provides a concise and clear definition of the Python programming language, highlighting its key features and its significance in the current landscape of software development. Python, renowned for its simple and readable syntax, has gained popularity due to its versatility and ease of use.

Objective: The primary objective of this study is to answer the question: What role does Python play in current software development and what are its future prospects? This analysis aims to understand the impact of Python on the software industry and assess how its evolution may influence future technological trends.

Methods: To address this objective, a literature review was conducted to gather relevant information about Python. Rigorous selection criteria were established to ensure the quality and timeliness of the consulted sources, and a data analysis was carried out that combined qualitative and quantitative approaches.

Results: The article summarizes the main advantages and disadvantages of Python, as well as exploring its various areas of application. Python offers advantages such as ease of learning and a robust ecosystem of libraries, but it also presents disadvantages such as slower execution speed compared to other languages.

Conclusions: In conclusion, the study synthesizes the results obtained, evaluates the current use of Python in software development, and provides practical recommendations for developers, highlighting the importance of staying updated with the latest versions and tools available.

Key words: Python, Programming language, Technological evolution, Simple syntax, Software development, Development tools.

INTRODUCCIÓN

Python es un lenguaje de programación interpretado, de alto nivel y orientado a objetos, que se destaca por su sintaxis simple y legible. Esto lo convierte en una herramienta accesible tanto para principiantes como para desarrolladores avanzados. Además, es multiparadigma, lo que permite a los programadores adoptar enfoques variados, como la programación funcional o imperativa (Romero Munizaga, 2022).

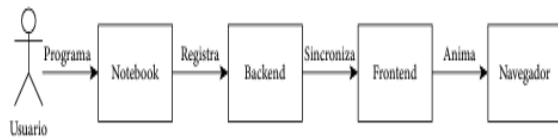


Figura 1. Flujo de la información

Desde su creación por Guido van Rossum en 1991, Python ha crecido exponencialmente en popularidad, impulsado por su versatilidad en áreas como desarrollo web, ciencia de datos, automatización y aprendizaje automático (Rawat, 2020). En la actualidad, es uno de los lenguajes más utilizados en la industria, especialmente en startups, tecnología y academia, debido a su amplia comunidad y vasto ecosistema de bibliotecas (Persson, 2021).

Este artículo tiene como objetivo analizar y evaluar el uso de Python en el desarrollo de software, destacando sus principales ventajas, como su facilidad de uso y la rapidez en el desarrollo, así como algunas desventajas, como su menor rendimiento en comparación con otros lenguajes más optimizados. También se explorarán las áreas de aplicación más comunes, como el análisis de datos, la inteligencia artificial y el desarrollo web.

La pregunta de investigación que guía este análisis es: ¿Qué papel juega Python en el desarrollo de software actual y cuáles son sus perspectivas futuras?

La programación, el arte de dar instrucciones a las máquinas, ha transformado radicalmente nuestra sociedad. Desde los dispositivos que usamos a diario hasta los sistemas que gestionan infraestructuras críticas, la programación está presente en todos los aspectos de nuestra vida. Como señala Hill, la capacidad de las máquinas para realizar cálculos complejos ha revolucionado sectores como el transporte, la salud y la administración. Sin embargo, los inicios de la programación no fueron tan sencillos (Hill, 2024).

Antes de 1954, los programadores enfrentaban desafíos formidables al trabajar con lenguajes de máquina y ensamblador. La creación de FORTRAN marcó un hito en la historia de la programación, al permitir a los desarrolladores escribir código de manera más legible y eficiente (Pietrini, 2024). Este avance, junto con otros desarrollos posteriores, sentó las bases para la explosión de aplicaciones informáticas que vemos hoy en día.

Comprender la historia de la programación nos permite apreciar los avances logrados y los desafíos superados. Además, nos proporciona una perspectiva valiosa para comprender las tendencias actuales y futuras de esta disciplina.

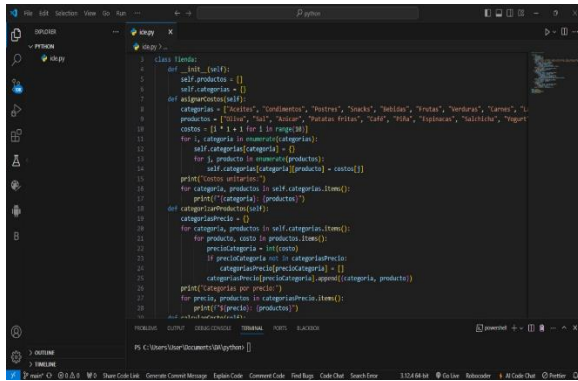


Figura 2. Editor de código (VS Code) mostrando un script de Python con una clase llamada "Tienda" que gestiona productos, categorías y costos.

ETAPAS DEL SOFTWARE:

El desarrollo de software ha experimentado una transformación radical desde sus inicios, pasando de ser un arte artesanal a una disciplina altamente compleja y sofisticada. En sus orígenes, el software era una mera adición al hardware, diseñado de manera personalizada para cada aplicación y con una distribución limitada (Morales-Carrillo, 2022). La falta de planificación y documentación, sumada a la baja movilidad laboral, dificultaba la gestión y el mantenimiento de los sistemas.

Con la llegada de los sistemas computacionales más complejos en los años 60 y 70, la interacción entre el humano y la máquina se intensificó. Esto dio lugar a aplicaciones más interactivas y a la necesidad de gestionar grandes volúmenes de datos. Sin embargo, el crecimiento exponencial del software y la creciente complejidad de los sistemas desencadenaron la llamada "crisis del software", caracterizada por retrasos en los proyectos, sobrecostos y baja calidad.

La década de los 70 marcó un punto de inflexión con la popularización de los microprocesadores y las

computadoras personales. El hardware se estandarizó rápidamente, mientras que el software se convirtió en un factor diferenciador (Higuera, 2024). La aparición de redes y sistemas operativos permitió el desarrollo de aplicaciones más sofisticadas y la interconexión de sistemas a nivel global.

En las últimas décadas, las tecnologías de la información han evolucionado a un ritmo vertiginoso. Internet, los sistemas operativos de red, las tecnologías orientadas a objetos y la inteligencia artificial han transformado radicalmente la forma en que desarrollamos y utilizamos el software. Hoy en día, el software está presente en prácticamente todos los aspectos de nuestra vida y continúa siendo un motor de innovación y cambio.

METODOLOGÍA

Revisión Bibliográfica: La búsqueda de información relevante sobre el uso de Python en el desarrollo de software se realizó a través de diversas fuentes académicas y profesionales. Se consultaron bases de datos especializadas como IEEE Xplore, Google Scholar y ACM Digital Library para acceder a artículos científicos y estudios de caso (Yasin, 2020). Además, se revisaron blogs especializados en tecnología y programación, como Real Python y Towards Data Science, para obtener perspectivas actuales y prácticas. La combinación de estas fuentes permitió obtener una visión completa y actualizada sobre el tema.

Criterios de Selección: Los estudios y fuentes de información fueron seleccionados con base en tres criterios principales: relevancia, actualidad y metodología. Se priorizaron estudios que ofrecieran una visión directa sobre la aplicación de Python en el

desarrollo de software y que fueran recientes, para asegurar que la información reflejara las tendencias actuales. También se consideró la metodología utilizada en cada estudio, buscando investigaciones que emplearan enfoques rigurosos y proporcionaran datos verificables (Contreras-Bravo, 2021).

Análisis de Datos: Los datos recopilados se analizaron mediante un enfoque mixto. Se llevó a cabo un análisis cualitativo para examinar las perspectivas y opiniones sobre el uso de Python, identificando tendencias y patrones en los comentarios de expertos y estudios de caso (Ruiz Lopez, 2021). Además, se realizó un análisis cuantitativo de las métricas disponibles, como estadísticas de uso y desempeño en diferentes aplicaciones. Este enfoque combinado permitió obtener una visión integral de las ventajas y desventajas de Python en el desarrollo de software.

RESULTADOS Y DISCUSIÓN

Python destaca por su facilidad de aprendizaje y uso, gracias a su sintaxis clara y concisa, lo que lo convierte en un lenguaje accesible tanto para principiantes como para programadores experimentados. La amplia comunidad y documentación disponible también facilitan la resolución de problemas y la adquisición de nuevas habilidades. Su versatilidad es otra de sus fortalezas, ya que se aplica en diversas áreas como desarrollo web, ciencia de datos, machine learning y automatización. El ecosistema de librerías de Python es robusto, con herramientas como NumPy y Pandas para análisis de datos, TensorFlow y PyTorch para machine learning, y muchas otras que amplían sus capacidades. Además, su portabilidad permite que Python funcione en múltiples plataformas, desde sistemas operativos Windows y Linux hasta macOS.

Desventajas de Python:

A pesar de sus muchas ventajas, Python presenta algunas desventajas. La velocidad de ejecución puede ser más baja en comparación con lenguajes compilados como C++ o Java, lo que lo hace menos adecuado para aplicaciones que requieren alta eficiencia en tiempo de ejecución. También puede tener un consumo de memoria relativamente alto, lo que puede ser una desventaja en entornos con limitaciones de recursos.

Áreas de Aplicación:

Python tiene una amplia gama de aplicaciones. En el desarrollo web, se utilizan frameworks como Django y Flask para construir aplicaciones robustas. En ciencia de datos y análisis, herramientas como NumPy, Pandas y Matplotlib son fundamentales para manipulación y visualización de datos. En el campo de machine learning, TensorFlow y PyTorch son ampliamente usados para desarrollar modelos avanzados. También se emplea en la automatización de tareas, permitiendo a los desarrolladores automatizar procesos repetitivos de manera eficiente. Además, Python se usa en el desarrollo de juegos con bibliotecas como Pygame.

Casos de Éxito:

Empresas como Google, Facebook y Spotify han utilizado Python en diversos aspectos de sus operaciones, desde la gestión de datos hasta la implementación de algoritmos de machine learning. Proyectos destacados incluyen el sistema de recomendación de Netflix y el análisis de datos de Instagram.

Comparación con Otros Lenguajes:

Comparado con lenguajes como Java, C++ y JavaScript, Python ofrece ventajas competitivas en

términos de facilidad de uso y desarrollo rápido debido a su sintaxis sencilla y a su extensa biblioteca estándar. Mientras que Java y C++ pueden ser más rápidos en ejecución y eficientes en el manejo de recursos, Python permite un desarrollo más ágil y flexible. JavaScript, aunque esencial para el desarrollo web, no tiene el mismo nivel de soporte para áreas como ciencia de datos y machine learning, donde Python es particularmente fuerte.

CONCLUSIONES

El estudio revela que Python es un lenguaje altamente valorado en el desarrollo de software debido a su sintaxis simple y su amplia versatilidad. Las principales ventajas identificadas incluyen su facilidad de aprendizaje, un ecosistema rico en librerías como NumPy y TensorFlow, y su portabilidad a múltiples plataformas. Sin embargo, también se han observado desventajas, como su menor velocidad de ejecución en comparación con lenguajes compilados y su mayor consumo de memoria. Python destaca en áreas como desarrollo web, ciencia de datos, machine learning y automatización, con casos de éxito notables en empresas líderes de tecnología.

Evaluación del Uso de Python:

Python ha tenido un impacto significativo en el desarrollo de software, ofreciendo una plataforma flexible y eficiente para una amplia gama de aplicaciones. Su capacidad para facilitar un desarrollo rápido y su amplia comunidad de soporte lo han consolidado como una herramienta clave en la industria. En el futuro, Python se espera que continúe evolucionando con mejoras en rendimiento y nuevas librerías, manteniéndose relevante en áreas

emergentes como la inteligencia artificial y la automatización.

Limitaciones del Estudio:

El estudio enfrenta algunas limitaciones, como la posible falta de representación de todas las áreas de aplicación y el enfoque limitado en ciertos tipos de estudios de caso. Además, la evolución constante del lenguaje y de las tecnologías asociadas puede afectar la vigencia de los hallazgos. Futuras investigaciones podrían explorar en mayor profundidad el rendimiento de Python en entornos específicos y comparaciones más detalladas con otros lenguajes emergentes.

RECOMENDACIONES

Para desarrolladores que desean utilizar Python:

Aprovechar la extensa documentación y recursos comunitarios para facilitar el aprendizaje y la resolución de problemas.

Considerar el uso de herramientas y librerías especializadas para optimizar el rendimiento en áreas como machine learning y análisis de datos.

Mantenerse al tanto de las actualizaciones y mejoras del lenguaje para utilizar las últimas características y técnicas.

Sugerencias para futuras investigaciones:

Investigar el impacto de las nuevas versiones de Python en el rendimiento y la eficiencia en aplicaciones específicas.

Realizar estudios comparativos más detallados entre Python y otros lenguajes emergentes en áreas como el desarrollo web y la ciencia de datos.

Explorar cómo las innovaciones en hardware y plataformas pueden influir en el uso de Python y su competitividad frente a otros lenguajes.

CONTRIBUCIONES DE LOS AUTORES:

Daniel Antonio Arias Vivanco – Introducción e investigación de fuentes.

Mariana Saray Defas Itaz – Desarrollo y aplicación de metodología.

Esteban David Jarrín Sánchez – Análisis de resultados, conclusiones y recomendaciones.

REFERENCIAS

- Romero Munizaga, J. L. (2022). Librería de visualización de estructuras de datos.
- Rawat, A. (2020). A Review on Python Programming. *International Journal of Research in Engineering, Science and Management*, 3(12), 8-11.
- Persson, I., & Khojasteh, J. (2021). Python packages for exploratory factor analysis. *Structural Equation Modeling: A Multidisciplinary Journal*, 28(6), 983-988.
- Hill, C., Du, L., Johnson, M., & McCullough, B. D. (2024). Comparing programming languages for data analytics: Accuracy of estimation in Python and R. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, e1531.
- Pietrini, R., Paolanti, M., & Frontoni, E. (2024, April). Bridging Eras: Transforming Fortran Legacies into Python with the Power of Large Language Models. In *2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI)* (pp. 1-5). IEEE.
- Morales-Carrillo, J., Cedeño-Valarezo, L., Bravo, J. S. C., & Calderón, J. G. O. (2022). Metodologías de desarrollo de software y su ámbito de aplicación: Una revisión sistemática. *Revista Ibérica de Sistemas e Tecnologías de Informação*, (E47), 29-45.
- Higuera, J. A. B., Quintero, J. M. R., & Araiza, C. A. R. (2024). Revisión sistemática de literatura de las actuales tecnologías de los microprocesadores. *Revista de Investigación en Tecnologías de la Información*, 12(25), 38-47.
- Yasin, A., Fatima, R., Wen, L., Afzal, W., Azhar, M., & Torkar, R. (2020). On using grey literature and google scholar in systematic literature reviews in software engineering. *IEEE access*, 8, 36226-36243.
- Ruiz Lopez, J. F. (2021). Revisión sistemática de publicaciones de ciencia de datos.
- Contreras-Bravo, L. E., Tarazona-Bermúdez, G. M., & Rodríguez-Molano, J. I. (2021). Tecnología y analítica del aprendizaje: una revisión a la literatura. *Revista científica*, (41), 150-168.