COMPUTATION E INFORMATICA

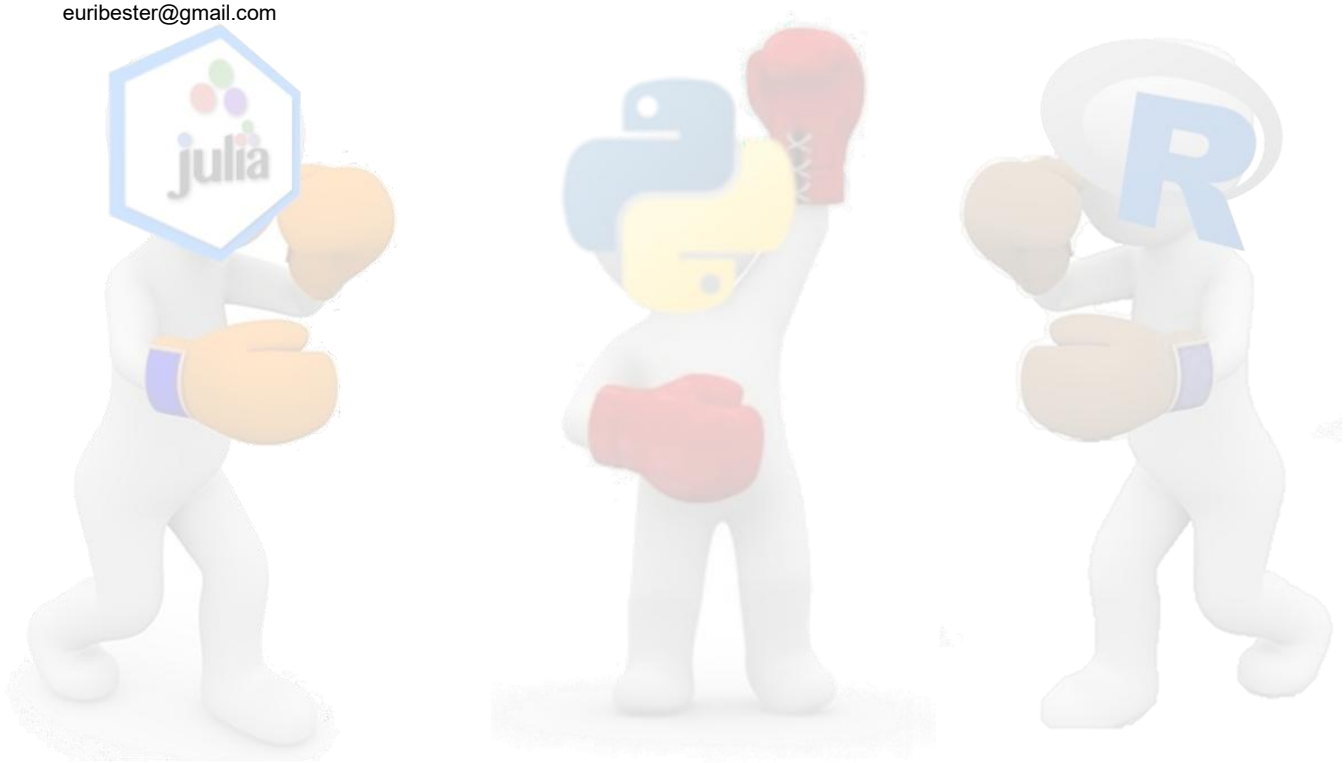
Recibido 14 May. 2025 *ReCIBE, Año 14 No. 3, Dic. 2025*

Aceptado 28 Jul. 2025

La batalla entre julia, python y r por el dominio en ciencia de datos y desarrollo

The competition among julia, python and r for leadership in data science and development

Jessica Mendoza
jessica.mendoza.j0610@gmail.com*
Aristeo Garrido Hernández[1]
agh@azc.uam.mx
Jesús Barrón Vidales
jbarronv@uttecamac.edu.mx
Roberto Efraín Uribe Capulín
euribester@gmail.com



[1] Universidad Autónoma Metropolitana

RESUMEN

Este estudio compara los lenguajes Julia, Python y R en el contexto de la ciencia de datos y el desarrollo de software. Evaluando rendimiento, bibliotecas, comunidad y licencias, mediante pruebas de velocidad —incluyendo operaciones matriciales, manipulación de datos y visualización— se observó que Julia es más rápido en cálculos numéricos, ideal para computación científica, mientras que Python ofrece equilibrio, es eficiente en procesamiento de datos y gráficos y tiene un ecosistema robusto y una gran comunidad en línea. Por su lado R sobresale en análisis estadístico y visualización, pero es más lento en tareas intensivas.

En el ámbito del desarrollo web, Python domina con frameworks maduros como Django y Flask, mientras que las alternativas en R (Shiny) y Julia (Genie) tienen un alcance limitado, además de la diferencia de licencias; Julia y Python utilizan MIT y PSF que son flexibles, adecuadas para proyectos comerciales, mientras que R emplea GPL que es más restrictiva.

Aunque cada lenguaje tiene fortalezas particulares, se observó observar que Python es versátil para proyectos generales, Julia destaca por su alto rendimiento y R sobresale en análisis estadístico. Por lo tanto, la elección del lenguaje depende del contexto y los objetivos del proyecto.

Palabras clave: Lenguajes de programación, Julia, Python, R, comparación, ciencia de datos, desarrollo de software.

ABSTRACT

This study compares Julia, Python, and R programming languages in the context of data science and software development. By evaluating performance, libraries, community support, and licensing, through conducting speed tests —including matrix operations, data manipulation, and visualization— it was observed that Julia is the fastest in numerical computations, making it ideal for scientific computing, while Python offers a balanced solution, it is efficient in data processing and graphics, and features a robust ecosystem and a large online community. R stands out in statistical analysis and visualization but is slower in computationally intensive tasks.

In the field of web development, Python leads with mature frameworks such as Django and Flask, while the alternatives in R (Shiny) and Julia (Genie) have more limited reach. Additionally, licensing differs: Julia and Python use MIT and PSF licenses, respectively, which are flexible and suitable for commercial projects, whereas R uses the GPL license, which is more restrictive.

Although each language has particular strengths, it was observed that Python is versatile for general projects, Julia stands out for its high performance and R excels in statistical analysis. Therefore, the choice of language depends on the context and objectives of the project.

Keywords: Programming languages, Julia, Python, R, comparison, data science, software development.

1. INTRODUCCIÓN

La revolución digital, impulsada por lenguajes de programación como Python, Julia y R, ha transformado radicalmente nuestra comprensión y análisis de datos. Estos tres lenguajes de programación no solo han acelerado el progreso en inteligencia artificial, biomedicina y negocios, sino que también han puesto poderosas herramientas al alcance de científicos, ingenieros y analistas de todo el mundo. Las necesidades en programación son amplias y varían según el área de especialización y el contexto temporal. Por ello, es fundamental presentar las características elementales de estos tres lenguajes de programación, con el objetivo de facilitar la elección del lenguaje que mejor se adapte a la tarea específica a realizar.

La sintaxis accesible y el vasto ecosistema de Python lo convierten en un pilar esencial para los desarrolladores de todos los niveles. Según el sitio web Python (Welcome To Python.org, 2024), millones de personas en todo el mundo utilizan el lenguaje debido a su facilidad de uso y versatilidad, un ejemplo reciente de su uso en las nuevas tecnologías es el que tuvo en en el desarrollo de modelos de inteligencia artificial de gran escala como GPT-3 y GPT-4 de OpenAI, donde se incluyeron bibliiotecas de aprendizaje automatic de Python y sus bibliotecas para la creación y el despliegue de estas tecnologías (Brown et al., 2020).

Con una velocidad de ejecución comparable a la de C y capacidades matemáticas avanzadas, Julia redefine los límites de la informática científica (Julia Documentation. The Julia Language, 2024). Eschle et al. (2023), en su artículo publicado en la revista *"Computing And Software For Big Science"*, destaca cómo Julia permite a los investigadores realizar cálculos complejos a velocidades increíbles, permitiendo descubrimientos científicos antes impensables, tal es el caso del proyecto Earthscope, donde Julia se utiliza para procesar grandes volúmenes de datos sísmicos y geodésicos (Nowacki, 2020).

Mientras tanto, R sigue siendo la opción académica preferida para el análisis estadístico y la visualización de datos, gracias a sus ricos paquetes especializados y su profundo impacto en la comunidad científica, (R: The R Foundation, 2024). Tierney (2012), afirma que la capacidad de R para procesar datos complejos y producir visualizaciones de alta calidad lo ha convertido en el estándar de facto en muchos campos científicos, además R cuenta con su propio ecosistema CRAN (The Comprehensive R Archive Network), que es la red que alberga y distribuye todos los paquetes de R, demostrando la capacidad del lenguaje para gestionar una infraestructura de software compleja para la comunidad estadística.

Este trabajo presenta una comparación entre los lenguajes de programación R, Julia y Python, considerando características clave como la facilidad para el desarrollo web, disponibilidad de bibliotecas, comunidad y soporte, velocidad de procesamiento y aspectos de propiedad intelectual. El objetivo es ofrecer un análisis cuantitativo y cualitativo que facilite una toma de decisiones informada para el desarrollo de programas computacionales.

2. FACILIDAD EN EL DESARROLLO WEB: UN ANÁLISIS DE R, PYTHON Y JULIA

El desarrollo web ha evolucionado significativamente en los últimos años y los lenguajes de programación utilizados se han diversificado. Tradicionalmente, lenguajes como JavaScript y PHP han sido dominantes en esta área, pero cada vez se adoptan más lenguajes como Python, R y Julia. Aunque estos lenguajes fueron desarrollados originalmente para programación científica y análisis de datos, también han demostrado ser herramientas poderosas en el campo del desarrollo web., (Holovaty y Kaplan-Moss, 2009). Según Van Rossum y Drake (2011), este lenguaje presenta características de integración y una extensa biblioteca estándar que simplifica el desarrollo de aplicaciones complejas.

Conocido por su simplicidad y legibilidad, Python por ejemplo facilita la creación de aplicaciones web sólidas y escalables gracias a marcos como Django y Flask. Django es particularmente elogiado por su enfoque en promover la reutilización de componentes y prácticas de desarrollo limpias y eficientes. Según Van Rossum (2011), creador de Python, el lenguaje presenta características de integración y una extensa biblioteca estándar que simplifica el desarrollo de aplicaciones complejas.

Por su parte R se asocia más comúnmente con el análisis estadístico y la visualización de datos, pero también ha encontrado su lugar en el desarrollo web a través de paquetes como Shiny, que permite a los desarrolladores crear aplicaciones web interactivas, haciendo que las herramientas avanzadas de visualización y análisis sean accesibles para todos, incluso para los usuarios con poca experiencia en programación. Wickham (2016) destaca la capacidad de este paquete para transformar el análisis de datos en tiempo real en aplicaciones web accesibles, facilitando la difusión de resultados y la toma de decisiones basada en datos.

En el caso de Julia que es un lenguaje relativamente nuevo, ha logrado combinar la velocidad de los lenguajes compilados como C con la simplicidad de los lenguajes de alto nivel. Su diseño está optimizado para la informática numérica y de alto rendimiento, pero también se extiende al desarrollo web. Julia permite a los desarrolladores utilizar marcos como Genie para crear aplicaciones web rápidas y eficientes e integrar capacidades avanzadas de análisis de datos directamente en la web, (julialang.org,2024).

Si bien R y Julia ofrecen excelentes capacidades de análisis de datos y computación de alto rendimiento, Python es una mejor opción para el desarrollo web gracias a su simplicidad, legibilidad y ecosistema robusto de marcos como Django y Flask. Este lenguaje se destaca como la opción más versátil y eficiente. La flexibilidad de Python permite a los desarrolladores crear fácilmente aplicaciones web escalables y eficientes. Ha sido ampliamente adoptado en la industria tecnológica y está respaldado por la capacidad de integrarse con otras tecnologías.

3. DUELO DE BIBLIOTECAS: HERRAMIENTAS ESENCIALES EN R, JULIA Y PYTHON

Las bibliotecas y herramientas de lenguajes de programación especializados son esenciales para desarrollar e implementar soluciones complejas en diversos campos, incluida la ciencia de datos y el desarrollo web La popularidad de los lenguajes de programación R, Julia y Python está aumentando debido a su capacidad para simplificar tareas complejas.

Estos tres lenguajes han sido ampliamente utilizados para el análisis y ciencia de datos, cada uno tiene su propia biblioteca de herramientas de visualización de datos y kits de herramientas especializados. R ofrece ggplot2 que crea gráficos sofisticados y altamente personalizables utilizando una gramática gráfica consistente (Wickham, 2016). Plots.jl permite la creación de gráficos complejos y personalizables se incluye con Julia (Bezanson et al., 2017). Por otro lado, Python tiene Matplotlib, que ofrece una amplia gama de funciones para crear gráficos de alta calidad y es similar a ggplot2 en términos de flexibilidad y capacidad de personalización (Hunter, 2007).

En cuanto a la manipulación de datos, R incluye dplyr, que facilita esta tarea con una sintaxis intuitiva para filtrar, seleccionar y transformar conjuntos de datos, mientras que tidyr se usa para limpiar y organizar datos, lo que permite transformarlos en formatos más útiles para el análisis. (Wickham et al., 2021; Wickham y Henry, 2021). DataFrames.jl es una herramienta eficiente para la manipulación y transformación de datos en estructuras de datos tabulares que es similar a la funcionalidad de dplyr y tidyr (Bezanson et al., 2017). Python contiene una poderosa biblioteca llamada Pandas que permite operaciones eficientes de limpieza, transformación y agregación de datos. Su sintaxis intuitiva es similar a la de dplyr y tidyr de R (McKinney, 2010).

Además de potentes herramientas para visualización y manipulación de datos, R, Julia y Python también proporcionan bibliotecas especializadas para aplicaciones web interactivas y aprendizaje automático, Shiny permite a los usuarios crear aplicaciones web interactivas directamente, lo que permite compartir análisis de datos y visualizaciones interactivas (Chang et al, 2021), Dash.jl de Julia proporciona una funcionalidad similar a Shine que permite el desarrollo de aplicaciones web interactivas para compartir resultados de análisis de datos (Plotly Technologies Inc.,2019). Python cuenta con Streamlit, según su sitio web streamlist.io, (2024) actualmente esta herramienta se ha vuelto extremadamente popular debido a su enfoque en la simplicidad y facilidad de uso para crear rápidamente aplicaciones web interactivas, permite que desarrolladores y científicos de datos creen interfaces de usuario interactivas para visualizar datos, realizar análisis y compartir resultados de manera efectiva.

Kuhn, (2021), destaca la naturaleza única de R en el ámbito de machine learning con el uso de Caret, una biblioteca que proporciona una interfaz unificada para diversos métodos y técnicas de modelado para entrenar y evaluar modelos predictivos. Julia propone MLJ.jl, una herramienta de aprendizaje automático que ofrece una interfaz unificada parecida a caret, lo que permite la evaluación y comparación de varios modelos (Blaom et al, 2020). El ecosistema de aprendizaje automático de Python está respaldado por una biblioteca popular, scikit-learn, que proporciona un conjunto completo de algoritmos y herramientas para entrenar y evaluar modelos predictivos, según Pedregosa et al, (2011).

En la tabla 1 se muestra las comparaciones entre las bibliotecas para machine learning, desarrollo web y creación de gráficas interactivas para cada lenguaje.

Tabla 1. Comparativa de las bibliotecas disponibles en cada lenguaje de programación (R, Python y Julia), organizadas según su capacidad para la visualización de gráficos, el aprendizaje automático y el desarrollo web. El ecosistema de bibliotecas de Python es más extenso en comparación con R y Julia. La

versatilidad de Python y su amplia gama de aplicaciones son notables, con ejemplos que incluyen desarrollo web con Django y flask (Holovaty et al, 2013; Grinberg, 2018), ciencia y análisis de datos con pandas (Mckinney, 2010), visualización con Matplotlib y Seaborn (Hunter, 2007; Waskom, 2021), y aprendizaje automático con scikit-learn y TensorFlow (pedregosa et al, 2011; Abadi et al, 2016). R es una herramienta poderosa para estadística y visualización con bibliotecas como ggplot2 y dplyr, pero no se usa tan ampliamente (Wickham et al, 2019). A pesar de ser reconocida por su competencia en cálculos numéricos intensivos y bibliotecas como jump para optimización (Dunning et al, 2017), julia aún no ha alcanzado la etapa de madurez y diversidad que se encuentra en el ecosistema de Python.

Lenguajes de programación	Visualización de gráficos	Machine Learning	Desarrollo web
R	ggplot2 plotly lattice ggvis Highcharter leaflet Esquisse	Caret RandomFores e1071 glmnet xgboost	Shiny Plumber Rook Rserve Httpuv fiery RestRserve OpenCPU
Phvton	Matplotlib Seaborn Plotly Bokeh Altair Pygal Pandas Plotnine	Numpy Pandas Matplotlib SciPy Scikit-learn TensorFlow Keras Pytorch Theano Scrapy	Django Flask PastAPI Tornado Pyramid Bottle Falcon web2py CherryPy Hug Sanic TurboGears Dash Aiohttp Quart
Julia	Plots Makie Gadfly VegaLite Gtk	Sparks Flux scikilearn. jl Mlbase TensorFlow Mocha Strada	Ginie HTTP Mux WebSockets Mustache Gumbo Bukdu LombScargle LightXML JSON LibPQ SQLite DataFrames

4. COMUNIDAD Y SOPORTE: ANALIZANDO EL APOYO EN R, JULIA Y PYTHON

La comunidad y el soporte son factores clave a la hora de elegir un lenguaje de programación, especialmente en áreas como análisis de datos y desarrollo web. Julia, R y Python, tienen sus propias fortalezas y características únicas en términos de la comunidad y el soporte que brindan.

Sin duda, uno de los lenguajes más utilizados es Python, que también cuenta con una comunidad bastante grande. Numerosos recursos, desde foros en línea y grupos de discusión hasta tutoriales y cursos en línea, han sido generados por la adopción masiva de Python en diversas industrias. Por ejemplo, Stack Overflow tiene una colección considerable de preguntas y respuestas relacionadas con Python, lo que hace que sea más fácil encontrar problemas comunes (Vasilescu et al., 2013). Python Software Foundation y conferencias como PyCon fomentan una comunidad que avanza continuamente en el desarrollo del lenguaje y las mejoras de la biblioteca.

También existe una fuerte comunidad para R, particularmente en los ámbitos académico y de investigación. El Comprehensive R Archive Network (CRAN) es un repositorio centralizado que proporciona una multitud de paquetes desarrollados por la comunidad para casi cualquier tipo de análisis estadístico y visualización de datos (Hornik, 2012). El entorno para la colaboración y el intercambio de conocimientos lo proporciona la R Foundation y conferencias como UseR. La comunidad R es conocida por ser acogedora y cooperativa, lo cual resulta bastante útil para nuevos usuarios y desarrolladores.

Aunque Julia es más nueva que Python y R, está ganando popularidad rápidamente, la comunidad de Julia es muy activa e incluye científicos, científicos de datos y desarrolladores que contribuyen periódicamente al lenguaje y sus paquetes. La plataforma JuliaHub y el repositorio de paquetes JuliaRegistries son recursos clave centrados en el desarrollo comunitario (Besançon et al., 2017).

Eventos como JuliaCon y las comunidades en línea Discourse y GitHub son importantes para apoyar y colaborar entre los usuarios de Julia.

Según la clasificación de PYPL, Python ocupa el primer lugar en popularidad, R el sexto y Julia el 23 (pypl.github.io, 2025). Python también encabeza la clasificación de Tiobe, seguido de R en el puesto 14 y Julia en el 31 (tiobe.com, 2025). De manera similar, en el popular sitio web Stack Overflow, Python encabeza la lista en términos de popularidad, superando con creces a Julia y R. Esta tendencia también se refleja en el sitio web de RedMonk (Stack Overflow, 2024; O'Grady, 2024). Estas clasificaciones resaltan el dominio de Python como el más preferido y ampliamente utilizado de los tres lenguajes, distinguiéndose por su versatilidad, vasta comunidad y ricos recursos (Figura 1).

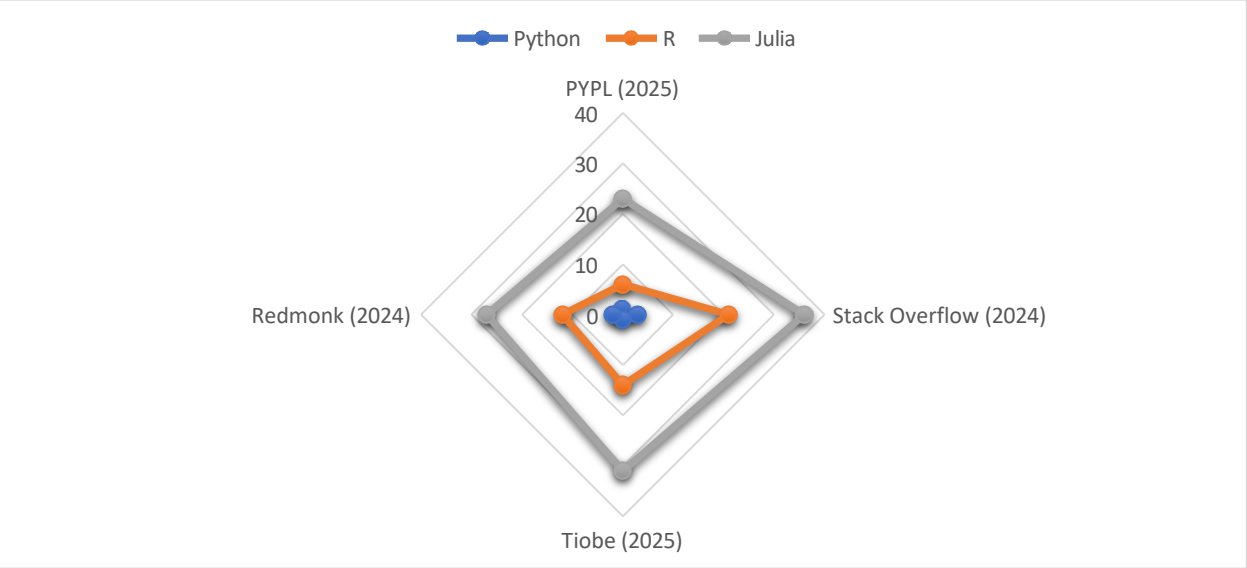


Figura 1. Orden de popularidad de cada lenguaje de programación, basado en datos de PYPL, TIOBE, Stack Overflow y RedMonk.

5. EL DESAFÍO DE LA VELOCIDAD: UN CARA A CARA ENTRE R, JULIA Y PYTHON

Como parte de una revisión sobre las ventajas y desventajas de los lenguajes de programación Python, Julia y R, orientada a la selección de la plataforma más adecuada para un desarrollo computacional, se presentan en este trabajo datos comparativos de velocidad de procesamiento. Esta información contribuye a la validación del proyecto de maestría titulado "Diseño y desarrollo de una plataforma web para aplicación en la técnica de espectroscopia infrarroja", se llevaron a cabo varias pruebas para evaluar los tiempos de respuesta de tres lenguajes de programación: R, Julia y Python. El objetivo de estas pruebas y los resultados obtenidos se detallan a continuación:

• **Agregación de dataframes:** Esta prueba evalúa la eficiencia y flexibilidad de cada software en el manejo de conjuntos de datos estructurados. Esta prueba nos permite analizar la capacidad de cada software para realizar operaciones como la suma, el promedio y la agrupación de datos en dataframes, esta prueba arroja información valiosa sobre su idoneidad para tareas de análisis de datos y ciencia de datos.

• **Desempeño sobre matrices:** Aquí se evalúa la eficiencia y velocidad con la que cada software puede manipular operaciones matriciales. Esta prueba nos proporcionaría información crucial sobre cómo cada software maneja tareas que requieren un procesamiento intensivo de datos estructurados en forma de matrices, como el análisis numérico, la simulación y la computación científica. Para esta evaluación se realizarán dos pruebas:

- **Modificación de elementos en una matriz:** A través de esta prueba obtenemos información sobre la capacidad de cada software para manipular y actualizar matrices de manera eficiente, lo que permite determinar cuál software es más adecuado para aplicaciones que requieren un procesamiento dinámico y ágil de datos en forma de matrices, como la optimización, la simulación y la manipulación de imágenes.
- **Operaciones con matrices:** Esta prueba facilita la comparación del rendimiento computacional, eficiencia en el uso de recursos y habilidad para paralelizar en tareas numéricas exigentes.

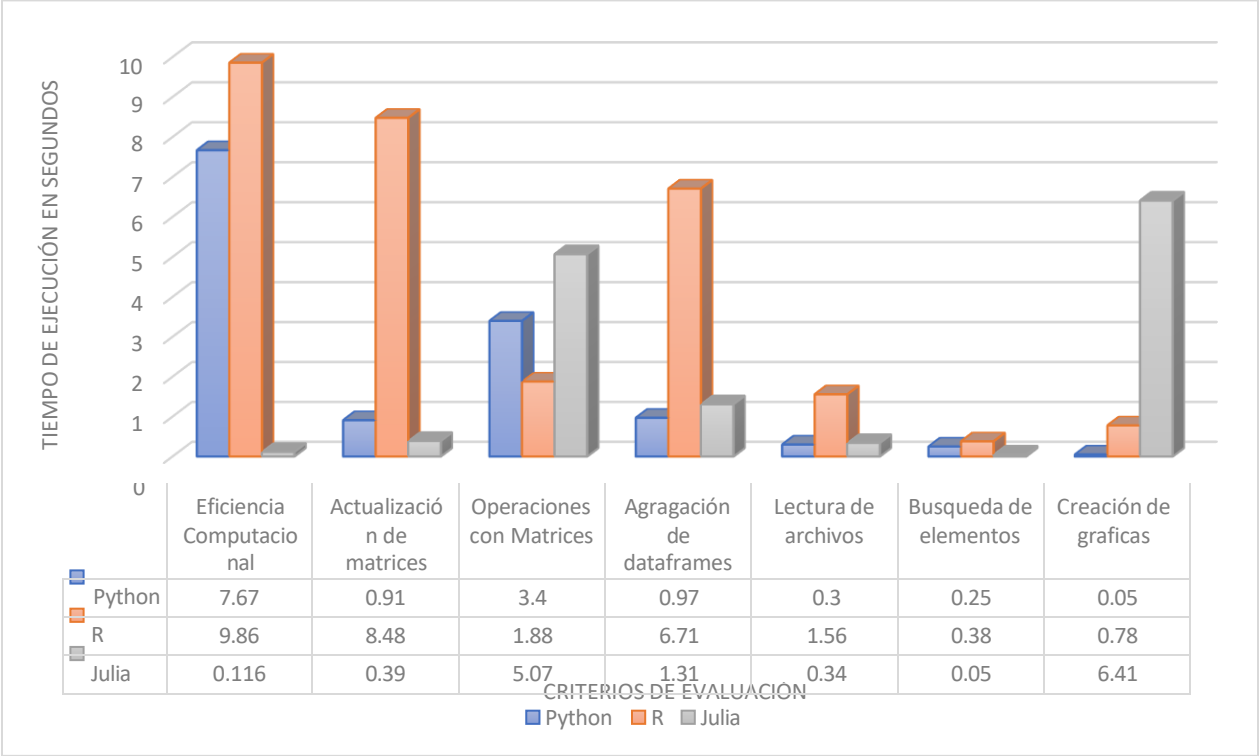
• **Creación de gráficas simples:** Se utiliza para evaluar la facilidad de uso y la eficacia con la que cada software puede generar visualizaciones básicas de datos. Esta prueba nos permitiría determinar qué software ofrece las herramientas más intuitivas y eficientes para crear gráficas simples, como diagramas de barras, gráficos de líneas y gráficos de dispersión. Además, nos ayudaría a identificar la rapidez con la que se puede generar y personalizar estas gráficas, lo que es fundamental para el análisis exploratorio de datos y la comunicación efectiva de resultados.

• **Eficiencia computacional:** Aquí se evalúa el rendimiento y la capacidad de cada software para realizar operaciones computacionales de manera rápida y eficiente. Esta prueba permite medir y comparar aspectos como el tiempo de ejecución, el uso de recursos del sistema y la escalabilidad, lo que proporciona una comprensión clara de cómo cada software maneja cargas de trabajo intensivas.

• **Búsqueda de elementos:** La finalidad de esta prueba es evaluar la eficiencia y la precisión con la que cada software puede encontrar elementos específicos dentro de conjuntos de datos lo que permite medir aspectos como el tiempo de búsqueda, la complejidad del algoritmo utilizado y la capacidad para manejar grandes volúmenes de datos.

• **Lectura de archivos:** Evalúa la eficiencia y la capacidad de cada software para procesar y manejar diferentes tipos de archivos de datos, en esta parte se mide el tiempo de lectura, la compatibilidad con diferentes formatos de archivo y la capacidad para manejar archivos de gran tamaño. La eficiencia en la lectura de archivos es esencial en muchas aplicaciones, como la importación de datos en análisis de datos, procesamiento de texto y manejo de archivos multimedia, lo que hace que esta prueba sea crítica para evaluar la utilidad y el rendimiento general de cada software.

Para cada una de las pruebas, se realizaron 100 ejecuciones por lenguaje de programación, registrándose los tiempos de ejecución correspondientes, se promediaron los tiempos de cada una de las pruebas, los resultados obtenidos se presentan en la Figura 2.



	Eficiencia Computacional	Actualización de matrices	Operaciones con Matrices	Agragación de dataframes	Lectura de archivos	Busqueda de elementos	Creación de graficas
Python	7.67	0.91	3.4	0.97	0.3	0.25	0.05
R	9.86	8.48	1.88	6.71	1.56	0.38	0.78
Julia	0.116	0.39	5.07	1.31	0.34	0.05	6.41

Figura 2. Tiempo promedio de ejecución de las pruebas, expresado en segundos, para cada lenguaje.

Julia se distingue por su alta eficiencia en tareas computacionales intensivas y en operaciones de búsqueda, mientras que Python muestra un mejor desempeño en la creación de gráficas y la agregación de dataframes, en contraste, R presenta un rendimiento significativamente inferior en la mayoría de las pruebas. El análisis de los tiempos promedio de ejecución confirma que Python y Julia ofrecen un nivel de eficiencia muy similar, superando ampliamente a R en términos generales (Figura 3).

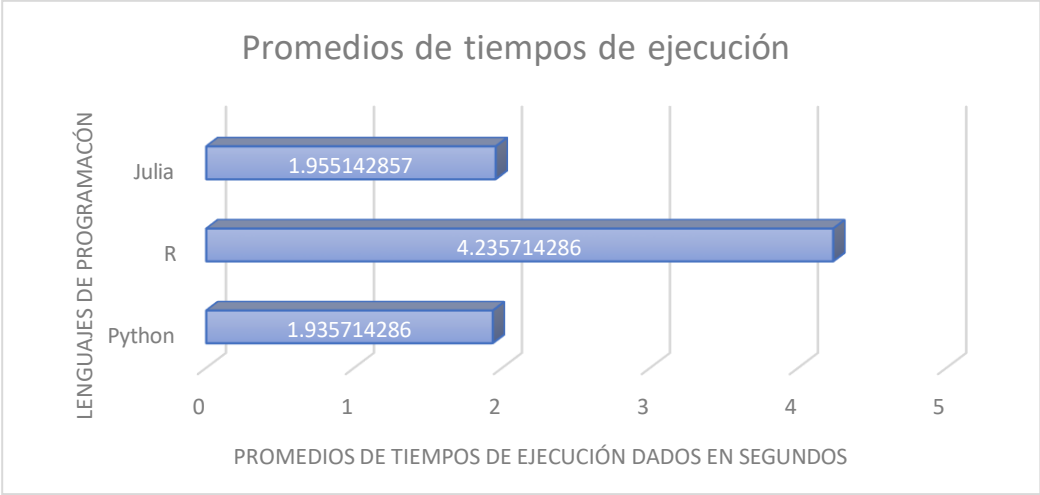


Figura 2. Promedios de tiempos de ejecución de las pruebas para cada lenguaje.

Aunque las pruebas reflejan una diferencia mínima en eficiencia entre Julia y Python, al considerar factores como el soporte comunitario y la disponibilidad de bibliotecas, Python se posiciona como la opción más sólida, su combinación de velocidad, amplia adopción y ecosistema robusto lo convierte en una de las herramientas más eficaces y preferidas por los desarrolladores, a pesar del crecimiento constante de las comunidades en torno a Julia y R.

6. PROPIEDAD INTELECTUAL Y LICENCIAS DE SOFTWARE EN R, JULIA Y PYTHON

La elección de un lenguaje de programación no se basa sólo en factores técnicos como el rendimiento, la facilidad de uso o la compatibilidad con otras herramientas y sistemas, sino también en las ramificaciones éticas y legales de su uso. Las licencias desempeñan un papel fundamental a este respecto, ya que definen los términos bajo los cuales el software puede utilizarse, modificarse y distribuirse. Las licencias pueden variar desde extremadamente restrictivas, que restringen el uso y la modificación del software, hasta extremadamente permisivas, que otorgan a los desarrolladores y usuarios una gran libertad, (Raymond, 2001).

El software de libre distribución, a menudo llamado software libre o de código abierto, se caracteriza por licencias que permiten a los usuarios no solo utilizar el software, sino también acceder al código fuente, modificarlo y redistribuirlo, esto promueve la transparencia, la colaboración y la innovación, ya que permite a una comunidad global de desarrolladores contribuir al desarrollo y mejora del software, (Stallman, 2021). La revisión de las licencias de software es, por tanto, un paso fundamental para cualquier organización o individuo que desee utilizar, desarrollar o contribuir a proyectos de software, asegurando que sus derechos y responsabilidades están claramente definidos y alineados con sus objetivos y valores, (Rosen, 2005).

Las licencias de software pueden tener implicaciones importantes para la propiedad intelectual. Un entendimiento claro de estas licencias ayuda a prevenir conflictos legales y asegura que los derechos de los autores y contribuyentes del software son respetados, (Fitzgerald y Bassett, 2003).

6.1 R

R es distribuido bajo la Licencia Pública General GNU (GPL), una licencia de software libre que permite a los usuarios ejecutar, examinar, compartir y alterar el software. La GPL estipula que cualquier trabajo derivado del software también debe distribuirse bajo la misma licencia, fomentando un entorno colaborativo y abriendo oportunidades para el desarrollo y mejoras de herramientas y paquetes (Free Program Foundation, 2007). Esta característica de la GPL es fundamental para mantener la libertad del software a lo largo de su ciclo de vida y asegurar que las mejoras y modificaciones beneficien a toda la comunidad de usuarios y desarrolladores (DiBona et al., 1999)

La naturaleza colaborativa de R bajo la GPL ha sido un factor clave en su adopción y mayor desarrollo en las comunidades científica y académica. La capacidad de modificar y mejorar el código permite a investigadores y estadísticos adaptar R a sus necesidades específicas, contribuyendo así al desarrollo de nuevos paquetes y extensiones que mejoran el ecosistema de R (Ihaka y Gentleman, 1996). Además, esta licencia promueve la transparencia en la investigación científica, ya que otros investigadores pueden revisar y reproducir los análisis y métodos. Esto es esencial para la reproducibilidad y validación de los resultados (Peng, 2011).

La amplia colección de paquetes disponibles en CRAN (Comprehensive R Archive Network), que facilita el intercambio y la distribución de herramientas desarrolladas por la comunidad, es otro ejemplo de cómo la GPL ha impactado el crecimiento de R (Gentleman et al., 2004).

6.2 JULIA

Julia usa la licencia MIT. Se trata de una licencia de software libre permisiva que permite a los usuarios hacer casi cualquier cosa con el código, siempre que se incluya una copia del aviso de derechos de autor y la licencia original. La licencia MIT es menos restrictiva que la GPL, lo que facilita la integración de Julia en otros proyectos de software y permite que más desarrolladores contribuyan al lenguaje sin preocuparse por las restricciones de la licencia (Licencia MIT, 1988). Esta flexibilidad ha llevado al rápido aumento de la popularidad de Julia, especialmente en campos como la ciencia de datos, la computación numérica y la inteligencia artificial (Bezanson et al., 2017).

La licencia MIT permite a los desarrolladores utilizar el código de Julia en sus propios proyectos sin tener que publicar el código fuente de esos proyectos, lo que podría ser un factor crucial para las empresas y los desarrolladores que buscan maximizar las capacidades de Julia sin comprometer su propiedad intelectual (Bezanson, Edelman, Karpinski, y Shah, 2017). La claridad y simplicidad de la licencia del MIT reducen las barreras legales y administrativas para la adopción y el desarrollo de Julia, fomentando una comunidad activa y en expansión (Perkel, 2019).

6.3 PYTHON

Python también se distribuye bajo la licencia Python Software Foundation License (PSF), que es similar a la licencia MIT. Esta licencia es permisiva y permite a los usuarios utilizar, modificar y redistribuir el software libremente. La flexibilidad de la licencia de Python ha contribuido a su amplia adopción y a la creación de una vasta cantidad de bibliotecas y frameworks de terceros (Python Software Foundation, 2001). Esta permisividad ha sido clave para que Python se convierta en uno de los lenguajes más populares y versátiles en la programación moderna (Van Rossum & Drake, 2009).

La licencia PSF permite a los desarrolladores integrar Python en aplicaciones comerciales sin tener que abrir el código fuente de la aplicación, lo que aumenta la aceptación de Python tanto en el mundo académico como en la industria (Lutz, 2013). La amplia disponibilidad de bibliotecas y marcos como Django para el desarrollo web, NumPy y SciPy para la informática científica y TensorFlow para el aprendizaje automático refleja la solidez del ecosistema Python, posible gracias a la naturaleza permisiva de la licencia (Ascher y Dubois, 1996).

Las diferencias en las licencias de estos lenguajes pueden influir en la elección de los desarrolladores y organizaciones en función de sus necesidades específicas y consideraciones legales. Por ejemplo, las empresas que desarrollan software propietario pueden preferir lenguajes con licencias permisivas como Julia y Python para evitar las obligaciones de compartir sus modificaciones bajo la misma licencia. Por otro lado, la comunidad académica y de investigación podría inclinarse hacia R debido a su enfoque en el software libre y abierto.

Las licencias de software de R, Julia y Python juegan un papel crucial en el desarrollo y adopción de estos lenguajes, afectando tanto a la comunidad de desarrolladores como a los usuarios finales. Al elegir un lenguaje de programación, es esencial considerar no solo las capacidades técnicas, sino también las implicaciones de propiedad intelectual y licencias que pueden influir en la viabilidad a largo plazo de los proyectos.

7. CONCLUSIÓN

En la batalla por el dominio en ciencia de datos y desarrollo, cada uno de los lenguajes R, Julia y Python tiene sus fortalezas y nichos específicos. R, con su fuerte enfoque en la estadística y el análisis de datos, se destaca en la academia y la investigación científica. Julia, con su rendimiento superior en cálculos numéricos y facilidad de uso, está ganando terreno rápidamente en la comunidad científica y de ingeniería.

Sin embargo, cuando se trata de desarrollo web, Python emerge indiscutiblemente como el líder. Su licencia permisiva, combinada con una vasta y robusta colección de bibliotecas y frameworks como Django y Flask, ha convertido a Python en la opción predilecta para desarrolladores de todo el mundo. La versatilidad de Python no solo lo hace ideal para la creación de aplicaciones web eficientes y escalables, sino que también permite una integración fluida con proyectos de ciencia de datos y aprendizaje automático, creando un ecosistema unificado y poderoso.

Python no es solo una herramienta, es el motor que impulsa la innovación en el desarrollo web moderno. Su capacidad para adaptarse a las demandas cambiantes del desarrollo tecnológico y su comunidad activa y en expansión aseguran que Python seguirá dominando en el futuro cercano. La elección es clara: para aquellos que buscan construir el próximo gran avance en el desarrollo web, Python es el camino para seguir. ¡El futuro del desarrollo está escrito en Python!

8. REFERENCIAS BIBLIOGRÁFICAS

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., . . . Zheng, X. (2016). TensorFlow: a system for large-scale machine learning. Operating Systems Design And Implementation, 265-283. https://doi.org/10.5555/3026877.3026899
- Ascher, D., & Dubois, P. F. (1996). "Numerical Python." Lawrence Livermore National Laboratory.
- Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). "Julia: A Fresh Approach to Numerical Computing." SIAM Review, 59(1), 65-98.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., . . . Amodei, D. (2020). Language Models are Few-Shot Learners. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2005.14165.
- DiBona, C., Ockman, S., & Stone, M. (1999). Open sources: Voices from the Open Source Revolution. O'Reilly Media, Inc.
- Dunning, I., Huchette, J., & Lubin, M. (2017). JuMP: A Modeling Language for Mathematical Optimization. SIAM Review, 59(2), 295-320. https://doi.org/10.1137/15m1020575
- Eschle, J., Gal, T., Giordano, M., Gras, P., Hegner, B., Heinrich, L., Acosta, U. H., Kluth, S., Ling, J., Mato, P., Mikhasenko, M., Briceño, A. M., Pivarski, J., Samaras-Tsakiris, K., Schulz, O., Stewart, G. . . A., Strube, J., & Vassilev, V. (2023). Potential of the Julia Programming Language for High Energy Physics Computing. Computing And Software For Big Science, 7(1). https://doi.org/10.1007/s41781-023-00104-x
- Fitzgerald, B., & Bassett, G. (2003). Legal Issues Relating to Free and Open Source Software. En Queensland University of Technology eBooks (p. 11). https://acuresearchbank.acu.edu.au/item/86381/legal-issues-relating-to-free-and-open-source-software
- Free software foundation. (2007). The GNU General Public License. https://www.gnu.org/licenses/gpl-3.0.html
- Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B.,

Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A. J., . . . Zhang, J. (2004). Bioconductor: open software development for computational biology and bioinformatics. GenomeBiology.com, 5(10), R80. https://doi.org/10.1186/gb-2004-5-10-r80

- Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python. O'Reilly Media.
- Holovaty, A., & Kaplan-Moss, J. (2009). The Definitive Guide to Django: Web Development Done Right (Expert's Voice in Web Development) (2.a ed.). Apress.
- Holovaty, A., & Kaplan-Moss, J. (2013). The Django Book Release 2.0. https://readthedocs.org/projects/djangobook/downloads/pdf/latest/
- Hornik, K. (2012). The comprehensive R Archive Network. Wiley Interdisciplinary Reviews. Computational Statistics, 4(4), 394-398. https://doi.org/10.1002/wics.1212
- Hunter, J. D. (2007c). MatPlotLib: a 2D Graphics environment. Computing In Science & Engineering, 9(3), 90-95. https://doi.org/10.1109/mcse.2007.55
- Ihaka, R., & Gentleman, R. (1996). R: A Language for Data Analysis and Graphics. Journal Of Computational And Graphical Statistics, 5(3), 299-314. https://doi.org/10.1080/10618600.1996.10474713
- Julia Documentation · The Julia Language. (2024). https://docs.julialang.org/en/v1/
- Líderes en formación tecnológica, reskilling y upskilling | OpenWebinars. (2024). OpenWebinars.net. https://openwebinars.net/blog/herramientas-y-librerias-de-machine-learning-en-python/
- Los mejores paquetes de visualización para R en 2023. (2023). BibTeX. https://bibtex.eu/es/blog/top-visualization-packages-for-r-in-2023/
- Lutz, M. (2013). "Learning Python." O'Reilly Media, Inc.
- McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings Of The Python In Science Conferences. https://doi.org/10.25080/majora-92bf1922-00a
- MIT License. (1988). Retrieved from https://opensource.org/licenses/MIT
- Nowacki, A. (2020). SeisModels.jl: A Julia package for models of the Earth's interior. The Journal Of Open Source Software, 5(47), 2043. https://doi.org/10.21105/joss.02043
- O'Grady, S. (2024). The RedMonk Programming Language Rankings: June 2024. https://redmonk.com/sogrady/2024/09/12/language-rankings-6-24/
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2012). Scikit-learn: Machine Learning in Python. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1201.0490
- Perkel, J. M. (2019). "Why Julia is Slowly Catching On." Nature, 572, 141-142.
- PYPL PopularitY of Programming Language index. (2025). https://pypl.github.io/PYPL.html
- Python Software Foundation. (2001). "Python Software Foundation License." Retrieved from https://www.python.org/psf/license/
- R: The R Foundation. (s. f.). https://www.r-project.org/foundation/
- Raymond, E. S. (2001). The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. «O'Reilly Media, Inc.»
- Roger D.Peng. Investigación reproducible en ciencias computacionales. Science 334 , 1226-1227 (2011). DOI: 10.1126/science.1213847
- Rosen, L. E. (2005). Open source licensing: Software Freedom and Intellectual Property Law. Prentice Hall.
- Stack overflow. (s. f.). https://survey.stackoverflow.co/
- Stallman, R. (2021). ¿Qué es el Software Libre? Free Software Foundation. https://www.gnu.org/philosophy/free-sw.html
- Streamlit. A faster way to build and share data apps, (2024). https://streamlit.io/
- Tierney, L. (2012). The R Statistical Computing environment. En Lecture notes in statistics

(pp. 435-447). https://doi.org/10.1007/978-1-4614-3520-4_41

- TIOBE Index - TIOBE. (2025). TIOBE. https://www.tiobe.com/tiobe-index/
- Van Rossum, G., y Drake, F. L. (2009). "The Python Language Reference Manual." Network Theory Ltd.
- Van Rossum, G., y Drake, F. L., Jr. (2011). An Introduction to Python. Network Theory. ISBN 978-1906966133
- Vasilescu, B., Serebrenik, A., y Devanbu, P. (2013). StackOverflow and GitHub: Associations between software development and crowdsourced knowledge. In 2013 International Conference on Social Computing (pp. 188-195). IEEE. https://doi.org/10.1109/SocialCom.2013.35
- Waskom, M. (2021b). seaborn: statistical data visualization. Journal Of Open Source Software, 6(60), 3021. https://doi.org/10.21105/joss.03021
- Wickham, H. (2016). ggplot2. En Use R! https://doi.org/10.1007/978-3-319-24277-4
- Wickham, H. (2021). Tidy Messy Data [R Package TidYR Version 1.1.3]. CRAN. http://ftp.udc.es/CRAN/web/packages/tidyr/index.html
- Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T., Miller, E., Bache, S., Müller, K., Ooms, J., Robinson, D., Seidel, D., Spinu, V., . . . Yutani, H. (2019). Welcome to the Tidyverse. Journal Of Open Source Software, 4(43), 1686. https://doi.org/10.21105/joss.01686
- Wickham, H., François, R., Henry, L., y Müller, K. (2021). A Grammar of Data Manipulation [R Package DPLyR Version 1.0.7]. CRAN http://cran.salud.gob.sv/web/packages/dplyr/index.html

Esta obra está bajo una licencia de Creative Commons
Atribución-NoComercial-Compartirlgual 4.0 Internacional.