

Análisis comparativo de IDEs enfocados a Machine Learning

Comparative analysis of IDEs focused on Machine Learning

José Roberto Sánchez Arteaga*
Empresa Municipal de Agua Potable y
Alcantarillado de Riobamba.
Riobamba-Ecuador
mvvthdp@hotmail.com
<https://orcid.org/0009-0009-9992-6252>

***Correspondencia:**
mvvthdp@hotmail.com

Cómo citar este artículo:
Sánchez, J. (2023). Análisis comparativo de
IDEs enfocados a Machine Learning. *Esprint*
Investigación, 2(2), 5-13.
<https://doi.org/10.61347/ei.v2i2.53>

Recibido: 1 de junio de 2023

Aceptado: 1 de julio de 2023

Publicado: 4 de julio de 2023

Resumen: El creciente uso de modelos de Machine Learning (ML) para gestionar grandes cantidades de datos, impulsado por la crisis de la Pandemia COVID-19, ha posibilitado el desarrollo productivo de los Entornos de Desarrollo Integrado (IDEs). Sin embargo, existen varios IDEs de desarrollo disponibles y elegir el más adecuado para tareas específicas puede ser un desafío. El objetivo de este estudio es proporcionar una revisión exhaustiva de las diversas herramientas de desarrollo integrado enfocadas para el campo de ML. Para ello, se diseñó una investigación comparativa donde se utilizaron fuentes científicas e índices de popularidad para determinar los lenguajes enfocados en ML, estos fueron Python y R. Posteriormente, a través de la documentación de cada IDE, se identificaron los siguientes: Spyder, PyCharm Community Edition, PyCharm Pro, DataSpell, R-Studio, RKWard, JupyterLab y Visual Studio Code que fue añadido por su popularidad y alta extensibilidad. En los resultados, se conceptualizan estos lenguajes e IDEs que determinamos para su comparación, y se llevó a cabo la comparación considerando características como su compatibilidad multiplataforma, capacidad de depuración con puntos de control, disponibilidad como código abierto, historiales de código, acceso a terminales, integración nativa de control de versiones, soporte para los lenguajes investigados, acceso a base de datos y compatibilidad con cuadernos Jupyter. En conclusión, esta investigación ofrece una comparación exhaustiva de IDEs considerando varias características útiles a la hora de seleccionar uno.

Palabras clave: Comparativa, desarrollo productivo, entornos de desarrollo integrado, gestión de datos, machine learning.

Abstract: The increasing use of Machine Learning (ML) models to manage large amounts of data, spurred by the Pandemic COVID-19 crisis, has enabled the productive development of Integrated Development Environments (IDEs). However, there are several development IDEs available and choosing the most suitable one for specific tasks can be challenging. The objective of this study is to provide a comprehensive review of the various integrated development tools focused for the ML field. For this purpose, a comparative research was designed where scientific sources and popularity indexes were used to determine the languages focused on ML, these were Python and R. Subsequently, through the documentation of each IDE, the following were identified: Spyder, PyCharm Community Edition, PyCharm Pro, DataSpell, R-Studio, RKWard, JupyterLab and Visual Studio Code which was added for its popularity and high extensibility. In the results, we conceptualize these languages and IDEs that we determined for comparison, and carried out the comparison considering features such as cross-platform compatibility, debugging capability with checkpoints, availability as open source, code histories, terminal access, native version control integration, support for the investigated languages, database access, and compatibility with Jupyter notebooks. In conclusion, this research provides a comprehensive comparison of IDEs considering several useful features when selecting one.

Copyright: Derechos de autor 2023 José Roberto Sánchez Arteaga.



Esta obra está bajo una licencia internacional
Creative Commons Atribución-
NoComercial 4.0.

Keywords: Benchmarking, data management, integrated development environments, machine learning, productive development.

1. Introducción

El empleo de modelos de Machine Learning (ML) para gestionar datos e información ha adquirido una importancia significativa, lo que ha incrementado la investigación académica en el campo de la IA en la última década, especialmente desde 2021. Este aumento se debe en parte a la crisis por la Pandemia COVID-19, que generó un flujo masivo de datos debido a la digitalización forzada durante la contingencia. En este ámbito, los Entornos de Desarrollo Integrado (IDE) desempeñan un papel crucial al proporcionar herramientas que simplifican la creación, entrenamiento y despliegue de estos modelos ML (Garcés-Giraldo et al., 2022).

Vargas (2023) define a los IDEs como un paquete de software que integra las herramientas de desarrollo en una única interfaz gráfica de usuario (GUI), de esta forma facilita la realización de tareas de desarrollo de software de manera más eficiente y rápida, aumentando la productividad. Las características más comunes entre los IDE y que apoyan de manera efectiva la productividad son:

- Editor de código: Facilita la escritura de código mediante el resaltado de la sintaxis, indicadores visuales, relleno automático y detección de errores (Red Hat, 2023).
- Compilación: Transforma el código de texto plano a un formato legible por el computador para su procesamiento (Vargas, 2023).
- Depurador: Según este autor opera en la fase de pruebas, facilita la detección de errores y, según AWS (2023) puede identificarlos en tiempo real.

Entre otras características adicionales se incluyen la refactorización y autocompletado de código, herramientas de automatización, gestión de versiones, y con frecuencia, proporcionan una amplia compatibilidad entre diversos lenguajes de programación, incluyendo los diseñados para facilitar la gestión de datos por ML.

Raschka & Mirjalili (2019) indican que actualmente se dispone de un recurso abundante: una gran cantidad de datos, tanto estructurados como no estructurados. El ML evolucionó como un subcampo de la Inteligencia Artificial (IA) y representa una alternativa eficiente para adquirir conocimiento a partir de estos datos. Esta evolución mejora gradualmente el rendimiento de los modelos predictivos y la toma de decisiones basadas en dichos datos. Bi et al. (2019) indican que su objetivo es el de permitir que las computadoras “aprendan” sin ser programadas directamente con el fin de mejorar su rendimiento en las tareas a través de la “experiencia”.

Esta experiencia se establece por la basta cantidad de datos de la que disponen. Como señala Manrique (2020), el ML resuelve situaciones de manera autónoma a partir del análisis de estos datos. En otras palabras, los resultados mejoran en función de la cantidad de datos disponibles. Además, se utilizan diversas técnicas y algoritmos que permiten reconocer patrones, extraer conocimiento, descubrir información y con ello realizar predicciones.

Con el tiempo, la mejora de los algoritmos de ML ha evolucionado este campo, y varios IDEs se han convertido en esa herramienta esencial para científicos de datos e ingenieros. Para comprender la relevancia de los IDEs es preciso explorar cómo estas herramientas han desempeñado un papel esencial en una serie de aplicaciones reales y cómo han contribuido a su éxito.

En el estudio de Gujral et al. (2017) se empleó R-Studio junto con el lenguaje R para detectar y predecir trastornos diabéticos, mediante SVM y Arboles de Decisión, logrando con éxito una precisión del 82 %, incluso, se propone su mejora mediante enfoques híbridos y lógica difusa. De la misma forma, Gutiérrez-Espinoza & Cabanillas-Carbonell (2021) destacan que, en su revisión sistemática la

Convolutional Neural Network (CNN) junto con R-Studio sobresale con mayor precisión en la detección temprana del cáncer de cuello uterino.

Por su parte, Devi et al. (2021) utilizan el aprendizaje automático y profundo aplicados a la clasificación de imágenes, destacando el uso de la CNN en la detección de objetos usando Tensorflow y Spyder, simplificando de esta forma el desarrollo de modelos de clasificación. En tanto, Israni (2019) también emplea Spyder para implementar los algoritmos: Máquina de Vectores de Soporte (SVM), K Vecinos más Cercanos (KNN) y Regresión Logística (LR) y obtiene una precisión del 92.78 % en el análisis del cáncer de mama para SVM.

A pesar de los avances logrados, todavía existen varios campos que pueden beneficiarse del ML y tienen a su disposición una gran variedad de IDEs. Tal es el caso de Saabith et al. (2021), que en su estudio analizan los IDEs empleados en ciencia de datos con Python, entre los que se encuentran: Jupyter Lab, Pycharm, Spyder, PyDev, Visual Studio, Thonny, Atom, Rodeo y AWS Cloud9. Asimismo, Gibson et al. (2020) mencionan además a Jupyter Notebook, entre las herramientas y técnicas usadas en su estudio. Sin embargo, existen otros IDEs disponibles y la elección del más adecuado para tareas específicas sigue siendo un desafío.

En consecuencia, el objetivo de la presente investigación se relaciona con proporcionar una revisión exhaustiva de las herramientas de desarrollo integrado enfocadas en el campo de ML. Esto se llevó a cabo al identificar los IDEs basados en los lenguajes que tuvieran este enfoque y al realizar una comparativa basada en características como su compatibilidad multiplataforma, capacidad de depuración con puntos de control, disponibilidad como código abierto, integración nativa de control de versiones, soporte nativo para los lenguajes investigados, acceso a base de datos, compatibilidad con cuadernos Jupyter, historiales de código y acceso a terminales.

2. Metodología

La metodología propuesta para la comparativa de IDEs enfocados a ML se basa en un enfoque de investigación comparativa. Los lenguajes de programación fueron seleccionados con base en fuentes científicas. Luego, con la ayuda de los índices de popularidad correspondientes al Top 20 de Tiobe (2023) y fuentes populares informales, se determinó que los lenguajes más populares enfocados a ML son Python que está en el primer lugar del top y R que tiene una alta probabilidad de entrar en el top. A continuación, se llevó a cabo una revisión de la literatura en la que se recopiló la información directamente de la documentación de los IDEs seleccionados.

La información recopilada incluyó datos relevantes sobre cada herramienta, de esta forma se establecieron las características de los IDEs que contienen funcionalidades específicas y detalladas de cada uno. Los datos recopilados se analizaron de manera comparativa. Con base en los resultados, se identificaron las características para seleccionar un IDE tanto para Python como para R.

3. Resultados

Lenguajes de programación enfocados en ML

En el ámbito de la Informática los lenguajes de programación tienen propósitos específicos, uno de los más relevantes según Manrique (2020) es la creación de aplicaciones de IA y ML. Considerando esta perspectiva teórica, los lenguajes de programación enfocados a ML son: Python, R, Julia y Matlab. Sin embargo, este estudio se centró únicamente en los lenguajes Python y R, que actualmente están en el índice de popularidad Tiobe (2023).

Y que según múltiples sitios populares informales reconocen a Python y R como los principales para el manejo de ML. A su vez, mencionan otros lenguajes, pero, es importante reconocer que estos no son adecuados como lenguaje principal para el desarrollo de ML, sin embargo, podrían ser usados como un complemento. A continuación, se conceptualizan los lenguajes seleccionados:

- **Python:** Según su documentación oficial en Python Software Foundation (2023), es un lenguaje de programación interpretado, orientado a objetos de alto nivel. Es conocido por su facilidad de uso, aprendizaje y mejora en la productividad. Se desarrolla bajo una licencia de código abierto y se puede distribuir gratuitamente, además, permite el uso de módulos y paquetes, fomentando la modularidad y reutilización del código.
- **R:** Según Manrique (2020) es un software especializado para análisis de datos, modelado estadístico y creación de gráficos de alta calidad. Posee una implementación similar al lenguaje S y es altamente extensible. Ofrece una amplia gama de técnicas estadísticas y gráficas, y se destaca por su facilidad para generar gráficos a nivel de publicación. Está disponible como software libre y es compatible con varias plataformas.

IDEs enfocados a ML

Una vez identificados los lenguajes que cubren las necesidades para el ML, es esencial encontrar los IDEs adecuados que soporten estos lenguajes. En ese sentido, se identificaron varios entornos compatibles con los lenguajes seleccionados, estos cumplirán el objetivo de aprovechar al máximo el potencial de Python y R, los IDEs investigados son:

- **Spyder:** Es un entorno de código abierto multiplataforma, desarrollado en Python para científicos, ingenieros y analistas de datos. Ofrece una combinación de funciones avanzadas de edición, análisis, depuración y creación de perfiles, haciéndola una herramienta de desarrollo integral. Además, cuenta con capacidades de exploración de datos, ejecución interactiva, inspección profunda y visualización detallada, todo en un solo paquete científico (Spyder Doc Contributor, 2023).
- **PyCharm:** Es un entorno multiplataforma que ofrece una variedad de herramientas esenciales gratuitas y de pago para los desarrolladores de Python. Está diseñado para crear un entorno de desarrollo eficiente para proyectos relacionados con la ciencia de datos, desarrollo web y programación en Python (JetBrains, 2023 A).
- **DataSpell:** Es una herramienta especializada en la realización de análisis de datos exploratorios y el prototipado de modelos de aprendizaje automático. Se encuentra diseñado para trabajar con Python y R (JetBrains, 2023 B).
- **RStudio:** Es un conjunto de herramientas diseñadas para mejorar la productividad en R y Python. Incluye una consola y un editor con resaltado de sintaxis que permite ejecutar código directamente. Además, proporciona herramientas para trazar, revisar el historial, depurar y gestionar su espacio de trabajo (Posit Software, 2023).
- **RKWord:** Es una herramienta basada en las bibliotecas de software KDE, utilizada para análisis estadísticos y gráficos con el lenguaje R. Está enfocado en ser accesible tanto para principiantes como para expertos en R, eliminando la necesidad de aprender R para principiantes y proporcionando funciones avanzadas para usuarios experimentados que necesiten escribir scripts en R. Su objetivo es fusionar flexibilidad y facilidad de uso en un mismo entorno. (Rödiger et al., 2012)
- **JupyterLab:** Es una interfaz web avanzada que ofrece flexibilidad y extensibilidad al permitir trabajos con cuadernos Jupyter, editores de texto, terminales y componentes personalizados de

manera integrada. Desde 2018, se encuentra en una fase experimental que busca reemplazar a Jupyter Notebook (Project Jupyter, 2018).

- **Visual Code:** Es un editor de código fuente potente, versátil, ligero y multiplataforma. Ofrece soporte integrado para los lenguajes principales: JavaScript, TypeScript y Node.js, además posee una amplia tienda de extensiones que lo hacen fácilmente configurable para otros lenguajes como C++, C#, Java, Python, PHP, Go, .NET. entre otros (Microsoft, 2023).

Comparativa de IDEs

En la tabla 1 se analizan algunos de los IDEs más populares en el contexto de ML con Python, considerando características como su compatibilidad multiplataforma, capacidad de depuración con puntos de control, disponibilidad como código abierto, historiales de código, acceso a terminales, integración nativa de control de versiones, soporte para los lenguajes investigados, acceso a base de datos y compatibilidad con cuadernos Jupyter.

Tabla 1.

Comparativa de IDEs que soportan Python

Características	Spyder	PyCharm Community Edition	PyCharm Pro	DataSpell
Compatibilidad Multiplataforma	SÍ	SÍ	SÍ	SÍ
Capacidad de depuración con puntos de control	SÍ	SÍ	SÍ	SÍ
Disponibilidad como código abierto	SÍ	SÍ	NO	NO
Historiales	De código y comandos	De código	De código y de consultas ejecutadas	De código y de consultas ejecutadas
Acceso a terminales	Gestión de múltiples consolas IPython, Consola integrada del SO	Gestión de múltiples consolas Python y consola integrada del SO	Gestión de múltiples consolas Python y consola integrada del SO	Gestión de múltiples consolas Python y consola integrada del SO
Control de versiones (nativo)	NO	SÍ	SÍ	SÍ
Lenguajes soportados (nativamente)	Python,	Python, HTML, XML, JSON, YAML, XSLT, XPath, Markdown.	Python, Cython, JavaScript, TypeScript, SQL, HTML/CSS, XML, JSON, YAML, XSLT, XPath, Markdown, compatibilidad con frameworks (Angular, Next.js, entre otros)	Python, R (Soporte básico), SQL, HTML, XML, JSON, YAML, XSLT, XPath, Markdown, Latex (en progreso)
Acceso a base de datos (nativo)	NO	NO	SÍ	SÍ
Compatibilidad con cuadernos Jupyter (nativo)	SÍ (funciones básicas de creación y edición)	NO	SÍ	SÍ

Las herramientas analizadas incluyeron tanto opciones de acceso abierto (Spyder y PyCharm Community Edition), como opciones de pago (Pycharm Pro y DataSpell). En ambos casos, se encontraron características comunes como la capacidad de ser multiplataforma, permitir la depuración

con el uso de puntos de interrupción, soporte integrado para Python y la mayoría integra un controlador de versiones.

Es importante destacar que las herramientas de pago incorporan de forma nativa el acceso a bases de datos y soporte para una amplia variedad de lenguajes, debido a que también están orientados al desarrollo web u otras áreas. Todas estas herramientas almacenan el código y algunas los comandos o consultas utilizadas, lo que facilita su recuperación, además, cuentan con consolas múltiples tanto para Python como para las terminales del Sistema Operativo (SO).

En la tabla 2 se analizan los IDEs destinados al uso del lenguaje R. Se ha incluido JupyterLab, que es compatible con Python y R. También se añadió Visual Studio Code (VS Code) debido a su popularidad entre los desarrolladores. Esta popularidad está respaldada por las encuestas de Stack Exchange (2023), que indican que sigue siendo la opción preferida de los desarrolladores. Cabe destacar que, aunque VS Code ofrece soporte para Python y R, esta funcionalidad requiere una configuración considerable.

Tabla 2.

Comparativa de IDEs que soportan R, Python y otros.

Características	R-STUDIO	RKWARD	JupyterLab	Visual Studio Code
Compatibilidad Multiplataforma	SÍ	SÍ	SÍ	SÍ
Capacidad de depuración con puntos de control	SÍ	NO	SÍ	SÍ
Disponibilidad como código abierto	SÍ	SÍ	SÍ	SÍ
Historiales	De comandos	De comandos (command log)	Del código ejecutado en la consola del kernel	De consola ejecutados en la terminal
Acceso a Terminales	Consola R integrada y proporciona acceso al Shell del sistema a través de múltiples terminales	Consola R integrada	Consola disponible de acuerdo con el Kernel activo	Consola integrada y proporciona acceso al Shell del sistema a través de múltiples terminales
Control de versiones (nativo)	SÍ	NO	NO	SÍ
Lenguajes soportados (nativamente)	R	R	Python (predeterminado), además admite Kernel de: R, C++, Julia, Sage, Scala, entre otros.	JavaScript/TypeScript y Node.js
Acceso a base de datos (nativo)	NO	NO	NO	NO
Compatibilidad con cuadernos Jupyter (nativo)	NO	NO	SÍ	NO

En este caso, se analizaron IDEs de código abierto tanto de escritorio como basados en web. En su totalidad, son multiplataforma y admiten la depuración de código con puntos de interrupción. Sin embargo, no todos tienen una integración nativa de control de versiones, como el caso de RKWard y JupyterLab. El lenguaje nativo para R-Studio y RKWard es el lenguaje R mientras que para JupyterLab se enfoca en Python, aunque ofrece la posibilidad de instalar otros kernels para la gestión de R, entre otros. Por otro lado, Visual Code proporciona compatibilidad con varios lenguajes, incluyendo Python y R a través de plugins.

Es evidente que ninguno de estos IDEs integra el acceso a bases de datos y compatibilidad con cuadernos de Jupyter. Todas estas herramientas al igual que las anteriores en la tabla 1 llevan un registro de los comandos o código usado, además de admitir el uso de la consola R o proporcionar acceso a múltiples terminales.

En general, algunos de estos IDEs no tienen ciertas herramientas específicas integradas de forma nativa. No obstante, es posible ampliar sus funcionalidades agregando complementos o plugins desarrollados por terceros, con el objetivo de personalizar y enriquecer su entorno de desarrollo según las necesidades del proyecto en el que ese esté trabajando.

4. Conclusiones

Se identificaron varios IDEs basados en lenguajes de programación con un enfoque en ML (Python y R), y se observó una amplia variedad de opciones disponibles en diversas plataformas. Estos IDEs abordan las características básicas de un entorno de desarrollo integrado, asegurando el acceso a tecnologías tanto para desarrolladores como para científicos de datos, independientemente de su nivel de conocimiento.

Al realizar una comparativa basada en características entre los IDEs enfocados en ML, existen varias opciones en cuanto al nivel de conocimiento requerido para utilizarlos. En primer lugar, están aquellos que admiten Python, caracterizados por su facilidad de uso y aprendizaje. En este caso, la elección se reduce principalmente a decidir si optar por una herramienta gratuita o de pago. Por sus características destacan versiones de pago como DataSpell, mientras que Pycharm Community Edition pudiera constituirse como la herramienta libre por excelencia, si se desea utilizar incluso para desarrollo web o Spyder si solo se necesita para herramientas de ML.

Por otro lado, para utilizar IDEs basados en el lenguaje R, es útil tener conocimientos estadísticos básicos o avanzados. Sin embargo, a pesar de que R no está especialmente diseñado para facilitar su aprendizaje, las opciones más recomendadas para personas sin experiencia es RKWard y JupyterLab. Estas herramientas están enfocadas en la facilidad de uso tanto para principiantes como para expertos.

Si bien algunos IDEs carecen de características integradas, se pueden enriquecer mediante plugins o componentes de terceros. Muchas de estas funcionalidades como control de versiones, soporte para otros lenguajes, acceso a base de datos y compatibilidad con cuadernos de Jupyter pueden ser complementadas de esta manera.

Referencias

- Amazon Web Services – AWS. (2023). *¿Qué es un IDE (entorno de desarrollo integrado)?* Amazon. <https://aws.amazon.com/es/what-is/ide/>
- Bi, Q., Goodman, K. E., Kaminsky, J., & Lessler, J. (2019). What is machine learning? A primer for the epidemiologist. *American Journal of Epidemiology*, 188(12), 2222-2239. <https://doi.org/10.1093/aje/kwz189>
- Devi, M. R. S., Kumar, V. V., & Sivakumar, P. (2021). A review of image classification and object detection on machine learning and deep learning techniques. In *2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 1-8). IEEE. <https://ieeexplore.ieee.org/abstract/document/9676141>
- Garcés-Giraldo, L. F., Benjumea-Arias, M., Cardona-Acevedo, S., Bermeo-Giraldo, C., Valencia-Arias, A., Patiño-Vanegas, C., ... & García, R. B. (2022). Uso de inteligencia artificial en gestión de la información: una revisión bibliométrica. *Revista Ibérica de Sistemas e Tecnologias de Informação*, (E54), 506-517. <https://www.proquest.com/openview/cd91b567d2231af1ab3b1838a62b3948/1?pq-origsite=gscholar&cbl=1006393>
- Gibson, S., Issac, B., Zhang, L., & Jacob, S. M. (2020). Detecting spam email with machine learning optimized with bio-inspired metaheuristic algorithms. *IEEE Access*, 8, 187914-187932. <https://doi.org/10.1109/ACCESS.2020.3030751>
- Gujral, S., Rathore, A. & Chauhan, S. (2017). Detecting and Predicting Diabetes Using Supervised Learning: An Approach towards Better Healthcare for Women. *International Journal of Advanced Research in Computer Science*, 8(5). <http://www.ijarcs.info./index.php/Ijarcs/article/view/3674>
- Gutierrez-Espinoza, S., & Cabanillas-Carbonell, M. (2021). Machine Learning Analysis for Cervical Cancer Prediction, a Systematic Review of the Literature. In *2021 International Conference on e-Health and Bioengineering (EHB)* (pp. 1-6). IEEE. <https://doi.org/10.1109/EHB52898.2021.9657567>
- Israni, P. (2019). Breast cancer diagnosis (BCD) model using machine learning. *International Journal of Innovative Technology and Exploring Engineering*, 8(10), 4456-4463. <http://doi.org/10.35940/ijitee.J9973.0881019>
- JetBrains. (2023 A). *Getting started*. JetBrains. <https://www.jetbrains.com/help/pycharm/getting-started.html>
- JetBrains. (2023 B). *Get started*. JetBrains. <https://www.jetbrains.com/help/dataspell/quick-start-guide.html>
- Manrique, E. M. (2020). Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo. *Revista Ibérica de Sistemas e Tecnologias de Informação*, (E28), 586-599. <https://www.risti.xyz/issues/ristie28.pdf>
- Microsoft. (2023). *Getting Started*. Visualstudio. <https://code.visualstudio.com/docs>
- Posit Software. (2023). *Download RStudio*. Posit. <https://posit.co/downloads/>
- Project Jupyter. (2018). *Overview*. Readthedocs. https://jupyterlab.readthedocs.io/en/1.2.x/getting_started/overview.html

- Python Software Foundation. (2023). *What is Python? Executive Summary*. Python. <https://www.python.org/doc/essays/blurb/>
- Raschka, S. & Mirjalili, V., (2019). Python machine learning. (2ª ed) Marcombo. <https://n9.cl/bf3om>
- Red Hat. (2023). *¿Qué es y para qué sirve un IDE?* RedHat. <https://www.redhat.com/es/topics/middleware/what-is-ide>
- Rödiger, S., Friedrichsmeier, T., Kapat, P., & Michalke, M. (2012). RKWard: a comprehensive graphical user interface and integrated development environment for statistical analysis with R. *Journal of Statistical Software*, 49(9), 1-34. <https://doi.org/10.18637/jss.v049.i09>
- Spyder Doc Contributor. (2023). *Welcome to Spyder's Documentation*. Spyder-ide. <https://docs.spyder-ide.org/current/index.html>
- Stack Exchange. (2023). *2023 Developer Survey*. Stackoverflow. [https://survey.stackoverflow.co/2023/#:~:text=boxes%20in%20that%20row.\)-Integrated%20development%20environment,-Visual%20Studio%20Code%20remains](https://survey.stackoverflow.co/2023/#:~:text=boxes%20in%20that%20row.)-Integrated%20development%20environment,-Visual%20Studio%20Code%20remains)
- Saabith, S., Vinothraj, T., & Fareez, M. (2021). A review on Python libraries and Ides for Data Science. *Int. J. Res. Eng. Sci*, 9(11), 36-53. <https://n9.cl/fgh2f>
- Tiobe. (2023). *TIOBE Index for June 2023*. Tiobe. <https://www.tiobe.com/tiobe-index/>
- Vargas, D. (2023). *¿Qué es un entorno de desarrollo y en qué se diferencia de un entorno de desarrollo integrado (IDE)?* Hostinger. <https://www.hostinger.es/tutoriales/que-es-un-entorno-de-desarrollo>

Transparencia

Conflicto de interés

El autor declara que no existen conflictos de interés que influyan en la objetividad de este estudio.

Fuente de financiamiento

No se recibieron fondos financieros de ninguna organización que pudiera tener interés en los resultados presentados.

Contribución de autoría

José Roberto Sánchez Arteaga: Conceptualización, metodología, software, validación, análisis formal, investigación, gestión de datos, visualización, redacción - preparación del borrador original, redacción - revisión y edición, financiamiento, administración del proyecto, recursos, supervisión.

El autor contribuyó activamente en el análisis de los resultados, revisión y aprobación del manuscrito final.