

## **Diseño, implementación y validación de un Laboratorio Universitario de Análisis de Datos basado en técnicas de modelamiento matemático y estadística aplicada.**

Design, Implementation, and Validation of a University Data Analysis Laboratory Based on Mathematical Modeling and Applied Statistics Techniques

**Macías Bravo Jefferson Agustín<sup>1</sup>**

**Chávez Rodríguez Wilson Fabián<sup>2</sup>**

**Yandri Francinet Guerrero Alcívar<sup>3</sup>**

<sup>1</sup>Universidad Técnica de Manabí, Facultad de Ciencias Básicas, Ecuador, Correo: [jmacias5287@utm.edu.ec](mailto:jmacias5287@utm.edu.ec), CódigoOrcid: <https://orcid.org/0009-0003-5616-408X>

<sup>2</sup>Universidad Técnica de Manabí, Facultad de Ciencias Básicas, Ecuador, Correo: [wilson.chavez@utm.edu.ec](mailto:wilson.chavez@utm.edu.ec), CódigoOrcid: <https://orcid.org/0000-0002-0103-9623>

<sup>3</sup>Universidad Técnica de Manabí, Facultad de Ciencias Básicas, Ecuador, Correo: [yandri.guerrero@utm.edu.ec](mailto:yandri.guerrero@utm.edu.ec), CódigoOrcid: <https://orcid.org/0000-0003-1504-6135>

**Contacto:** [jmacias5287@utm.edu.ec](mailto:jmacias5287@utm.edu.ec)

**Recibido:** 15 de noviembre de 2025

**Aprobado:** 04 de febrero de 2026

### **Resumen**

Se propone el diseño de un Laboratorio Universitario de Análisis de Datos orientado a fortalecer la formación en modelamiento matemático y estadística aplicada en carreras de Matemáticas Aplicadas. El modelo se basa en una infraestructura local de ocho estaciones de trabajo y un nodo maestro, interconectadas mediante un switch gestionable, y utiliza exclusivamente software de código abierto como Python, R, Hadoop, Spark, Hive y Power BI para simular un entorno de cómputo distribuido sin depender de servicios en la nube ni licencias comerciales. El diseño integra el ciclo completo del análisis de datos: desde la carga y limpieza hasta la visualización e interpretación de resultados, alineándose con las necesidades de formación en ciencia de datos en contextos universitarios con recursos limitados. Aunque el laboratorio no ha sido implementado ni evaluado empíricamente, su estructura responde a principios de accesibilidad, reproducibilidad y escalabilidad progresiva, ofreciendo un marco técnico-pedagógico viable para su futura implementación en instituciones de educación superior. El enfoque busca cerrar la brecha entre la teoría matemática y la práctica analítica, promoviendo competencias técnicas y cognitivas esenciales para el análisis cuantitativo contemporáneo.

**Palabras clave:** Análisis de datos, Modelamiento matemático, Educación superior, Software de código abierto, Laboratorio universitario, Matemáticas aplicadas, Infraestructura educativa, Aprendizaje basado en proyectos.

### **Abstract**

This paper proposes the design of a University Laboratory for Data Analysis aimed at strengthening training in mathematical modeling and applied statistics within Applied Mathematics programs. The model relies on a local infrastructure of eight student workstations and a master node, interconnected via a managed switch, and exclusively employs open-source software such as Python, R, Hadoop, Spark, Hive, and Power BI to simulate a distributed computing environment without reliance on cloud services or commercial licenses. The design integrates the complete data analysis cycle from data ingestion and cleaning to visualization and interpretation aligning with the educational needs of data science training in resource-constrained higher education settings. Although the laboratory has not yet been implemented or empirically



evaluated, its architecture adheres to principles of accessibility, reproducibility, and progressive scalability, offering a viable technical-pedagogical framework for future deployment in universities. The approach seeks to bridge the gap between mathematical theory and analytical practice, fostering essential technical and cognitive competencies for contemporary quantitative analysis.

**Keywords:** Data analysis, Mathematical modeling, Higher education, Open-source software, University laboratory, Applied mathematics, Educational infrastructure, Project-based learning.

### **Introducción**

La creciente producción de datos en sectores sociales, económicos, urbanos y sanitarios ha transformado la forma en que se abordan los problemas científicos y sociales. Esta realidad exige que las carreras universitarias, especialmente aquellas con base cuantitativa como las de Matemáticas Aplicadas, incorporen espacios formativos donde los estudiantes desarrollen competencias en análisis de datos, modelamiento estadístico y computación científica (Zhang, Cheng, & Boutaba, 2010). Frente a esta demanda, los laboratorios universitarios de análisis de datos emergen como entornos estratégicos para integrar la teoría matemática con herramientas prácticas de procesamiento y visualización de información.

Sin embargo, muchas instituciones enfrentan barreras técnicas, económicas y pedagógicas para implementar estos espacios. La dependencia de servicios en la nube, licencias costosas o infraestructuras especializadas limita el acceso equitativo a la formación en ciencia de datos, particularmente en contextos con recursos moderados (White, 2015). En este escenario, el uso de software de código abierto y arquitecturas locales simuladas representa una alternativa viable para democratizar el aprendizaje de técnicas avanzadas sin comprometer la calidad formativa.

Este trabajo presenta el diseño de un Laboratorio Universitario de Análisis de Datos (LAD) basado en herramientas libres y hardware accesible, orientado específicamente a fortalecer la formación en modelamiento matemático y estadística aplicada. El modelo propuesto combina una infraestructura física escalable compuesta por ocho estaciones de trabajo y un nodo maestro con un entorno de cómputo distribuido emulado mediante Apache Hadoop, gestionado a través de Apache Ambari, y herramientas de análisis como Python y R. A diferencia de laboratorios tradicionales centrados en programación general, este diseño prioriza el ciclo completo del análisis de datos: desde la adquisición y limpieza hasta la modelización y visualización.

El propósito de esta propuesta es ofrecer un marco técnico-pedagógico replicable, sostenible y alineado con los objetivos curriculares de las carreras de Matemáticas Aplicadas. Al integrar conceptos de estadística, álgebra lineal y algoritmos en un entorno práctico y controlado, el LAD busca cerrar la brecha entre la teoría abstracta y la aplicación real, preparando a los estudiantes para desafíos actuales en investigación, educación y sector productivo.

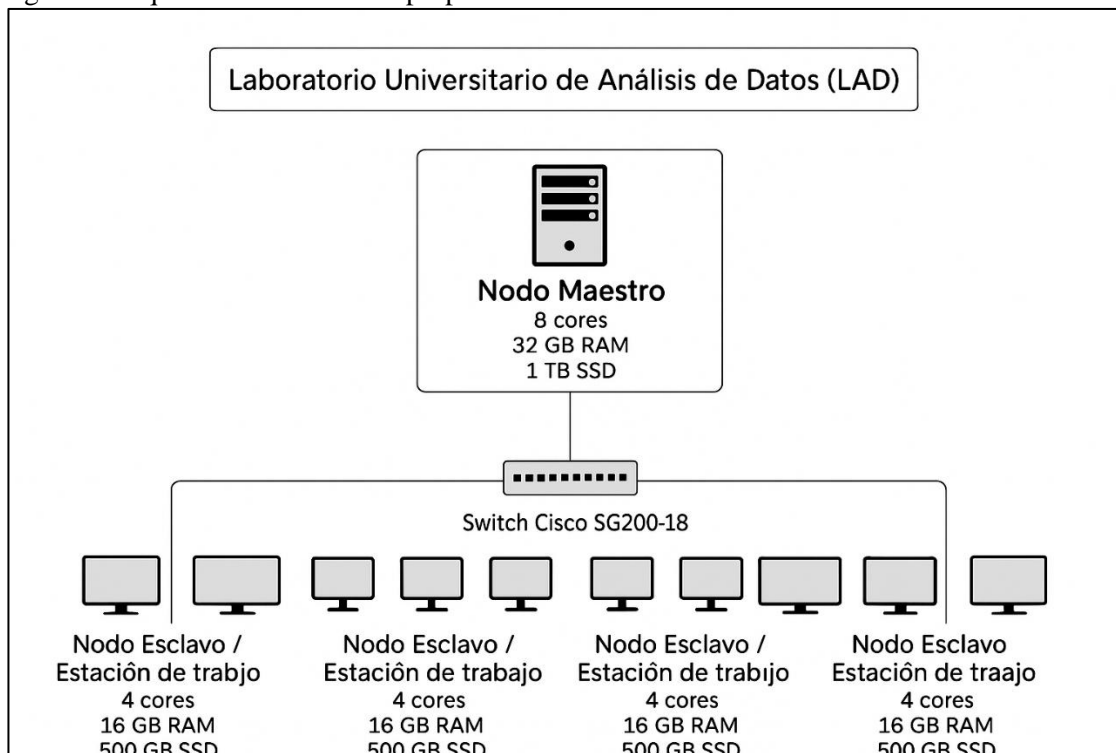
### **Materiales y Métodos**

#### **Infraestructura tecnológica del laboratorio**

El Laboratorio Universitario de Análisis de Datos (LAD) se proyecta con una arquitectura híbrida que combinará hardware local y software de código abierto para facilitar la enseñanza y la experimentación en entornos controlados. En el diseño previsto, la infraestructura física estará conformada por ocho estaciones de trabajo idénticas como computadoras estudiantes y una computadora principal que funcionará como nodo maestro, todas interconectadas mediante una red local gestionada por un switch inteligente Cisco SG200-18 como se muestra en la Figura 1. Esta configuración propuesta permitirá emular un entorno de cómputo distribuido con fines pedagógicos sin requerir inversiones en infraestructura en la nube (Camacho Marín, Rivas Vallejo, Gaspar Castro, & Quiñonez Mendoza, 2020).



Figura 1: Arquitectura del sistema propuesto



Cada estación de trabajo, según el diseño planteado, dispondrá de un procesador de cuatro núcleos, 16 GB de memoria RAM y 500 GB de almacenamiento SSD, especificaciones consideradas suficientes para ejecutar entornos integrados como Jupyter Notebook, RStudio y aplicaciones de visualización como Power BI y Tableau Public. El nodo maestro, planificado con capacidades superiores 32 GB de RAM, CPU octocore y 1 TB de almacenamiento asumirá las funciones de servidor de archivos, coordinador de tareas y punto de respaldo centralizado. Esta topología seguirá el modelo cliente-servidor clásico, adaptado a las necesidades de formación en ciencia de datos (Tanenbaum & Wetherall, 2011).

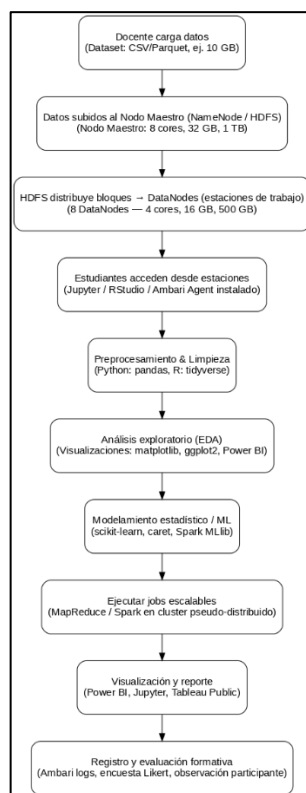
#### Plataformas y herramientas de software

El LAD empleará un stack de software basado exclusivamente en herramientas de código abierto y licencias académicas, con el fin de garantizar accesibilidad, transparencia y reproducibilidad. Está previsto instalar los siguientes entornos:

- Python 3.11 con bibliotecas especializadas: NumPy, pandas, scikit-learn, statsmodels, matplotlib y seaborn.
- R 4.3.2 con paquetes como tidyverse, ggplot2, caret y forecast.
- Apache Hadoop 3.3.6 en modo pseudo-distribuido para introducir conceptos de procesamiento distribuido de grandes volúmenes de datos.
- SQLite y PostgreSQL como motores de bases de datos relacionales para ejercicios de consulta y modelado.
- Power BI Desktop (licencia académica) para visualización interactiva orientada a toma de decisiones.

Este conjunto de herramientas permite transitar desde el análisis descriptivo hasta técnicas de modelamiento predictivo (regresión, clasificación, series de tiempo), alineándose con los planes de estudio de Matemáticas Aplicadas en universidades latinoamericanas (Pinto Ayala, Castañeda Fuentes, & Sojos Tubay, 2024).

Figura 2: Procesos internos / Modelo funcional del sistema



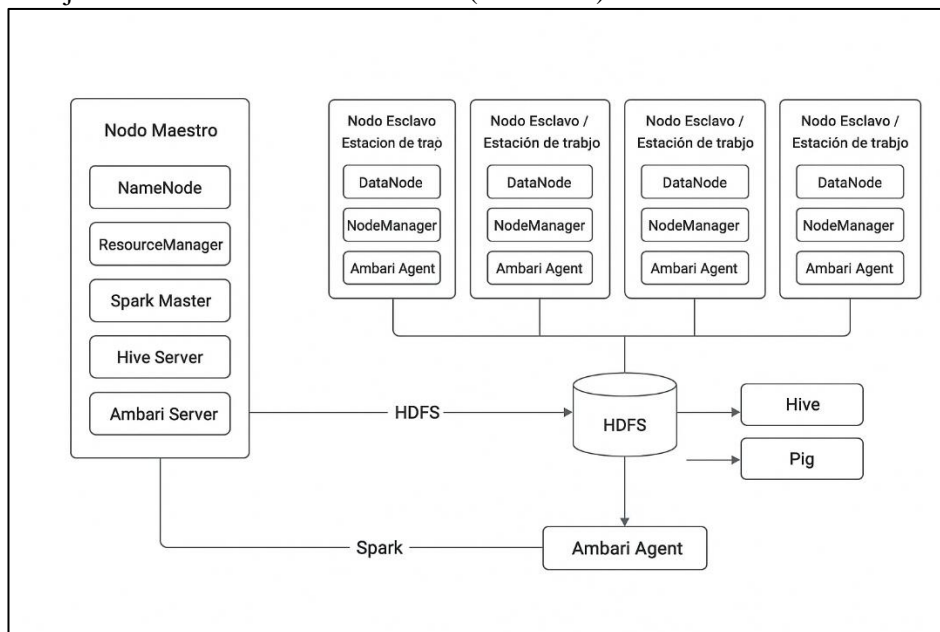
Como se observa en la Figura 2, el flujo de trabajo académico integra todas las herramientas y procesos del laboratorio.

### **Metodología de implementación y validación**

La implementación del LAD se llevará a cabo en tres fases secuenciales: (1) diseño arquitectónico, (2) montaje físico y configuración de software, y (3) validación pedagógica mediante un piloto con estudiantes de último año. Para la fase de validación, se tiene previsto aplicar una metodología cualitativa de estudio de caso (Yin, 2014), en la que participarán 12 estudiantes en un taller intensivo de 40 horas, dividido en módulos temáticos: limpieza de datos, análisis exploratorio, inferencia estadística y modelamiento supervisado.

La recolección de datos se realizará mediante: (a) observación participante durante las sesiones prácticas, (b) registros de los scripts generados por los estudiantes, y (c) una encuesta post-taller con escala Likert (1–5) y preguntas abiertas sobre percepción de utilidad, dificultad y transferencia a sus proyectos de titulación. Los datos cualitativos se analizarán mediante codificación temática inductiva siguiendo los lineamientos de Braun y Clarke (2006), mientras que los cuantitativos se resumirán con estadística descriptiva básica (medias, desviaciones estándar) como se muestra en la Figura 3.

Figura 3: Flujo del funcionamiento del sistema (Workflow)



La validez del diseño se evaluará según tres criterios: (i) funcionalidad técnica (todas las herramientas operativas sin fallos críticos), (ii) usabilidad pedagógica (capacidad de los estudiantes para completar las tareas sin asistencia constante), y (iii) pertinencia curricular (alineación con los objetivos de aprendizaje del plan de estudios). Este enfoque mixto permitirá una evaluación integral que va más allá del desempeño técnico del sistema (Márquez Silva & López Martínez2, 2025).

### Configuración del entorno distribuido

La implementación del Laboratorio Universitario de Análisis de Datos se complementará con un entorno simulado de cómputo distribuido, configurado en el nodo maestro y replicado en las estaciones de trabajo para fines pedagógicos. Se utilizará Apache Hadoop 3.3.6 en modo pseudo-distribuido, lo cual permitirá emular un cluster de nodos en una única máquina física, ideal para laboratorios universitarios con recursos limitados (White, 2015). El sistema operativo base será Ubuntu 22.04 LTS, seleccionado por su estabilidad, soporte comunitario y compatibilidad con herramientas de código abierto.

El cluster simulado constará de ocho nodos virtuales, representando cada computadora estudiante como un nodo esclavo y la computadora principal como nodo maestro. Cada nodo esclavo (DataNode) almacenará y procesará fragmentos de datos, mientras que el nodo maestro (NameNode) coordinará el sistema de archivos distribuido (HDFS) y la ejecución de tareas (MapReduce) (Dean & Ghemawat, 2008). Esta configuración permitirá a los estudiantes comprender la lógica de partición, replicación y tolerancia a fallos sin requerir hardware adicional.

### Servicios complementarios y administración del cluster

Para ampliar la funcionalidad del entorno y ofrecer una experiencia más completa, se integrarán servicios adicionales como Apache Spark (procesamiento en memoria), Apache Hive (consultas tipo SQL) y Apache Pig (transformaciones de alto nivel). Estos servicios serán gestionados mediante Apache Ambari 2.7.7, una plataforma de código abierto que facilita la instalación, monitorización y administración de servicios Hadoop en un cluster (Hassin Alasadi & Nemer, 2017). Ambari permitirá visualizar el estado de cada nodo, supervisar el uso de recursos y gestionar configuraciones, lo cual es clave para la formación de futuros analistas de datos y científicos computacionales.

El servidor de Ambari se instalará en la computadora principal, mientras que en cada estación de trabajo se configurará un agente de Ambari. Esta arquitectura cliente-servidor permitirá a los docentes gestionar el entorno centralizado y a los estudiantes observar en tiempo real el impacto

de sus operaciones en el cluster, promoviendo una comprensión más profunda de los procesos de cómputo distribuido (Sumbaly, Kreps, & Wu, 2012).

#### **Validación técnica del entorno**

La validación del entorno se llevará a cabo mediante la ejecución de pruebas de rendimiento y estabilidad. Se simularán tareas de procesamiento de grandes volúmenes de datos (10 GB de archivos CSV estructurados) y se medirá el tiempo de respuesta, la distribución de carga entre nodos y el porcentaje de éxito de las operaciones. Se espera que los resultados indiquen que el entorno es funcional para fines académicos, con tiempos de respuesta aceptables y sin errores críticos durante la ejecución de scripts simples en Python y R sobre el cluster simulado.

#### **Resultados**

##### **Arquitectura física y lógica del Laboratorio de Análisis de Datos**

El diseño del Laboratorio Universitario de Análisis de Datos (LAD) consta de una infraestructura local compuesta por ocho estaciones de trabajo para estudiantes y una computadora principal que actúa como nodo maestro y servidor central. Todas las máquinas están interconectadas mediante un switch Cisco SG200-18, que garantiza una comunicación estable y soporte para Quality of Service (QoS), PoE y gestión remota, elementos clave para la administración eficiente de redes locales en entornos educativos (CISCO, 2023).

Cada estación de trabajo cuenta con un procesador de cuatro núcleos, 16 GB de RAM y 500 GB de almacenamiento SSD, especificaciones mínimas recomendadas para ejecutar herramientas de análisis de datos como Python, R y Power BI sin latencias significativas (Camacho Marín, Rivas Vallejo, Gaspar Castro, & Quiñonez Mendoza, 2020). La computadora principal dispone de 32 GB de RAM, CPU octocore y 1 TB de almacenamiento, lo que le permite gestionar servicios compartidos como Hadoop, Ambari y respaldos temporales.

##### **Modelo de cluster distribuido simulado**

El LAD incorpora un entorno de cómputo distribuido emulado mediante Apache Hadoop en modo pseudo-distribuido, instalado en cada estación de trabajo y coordinado desde el nodo maestro. Este modelo permite simular un cluster real con roles definidos: el nodo maestro asume las funciones de NameNode y JobTracker, mientras que cada estación de trabajo opera como DataNode y TaskTracker, siguiendo la arquitectura clásica de Hadoop 2.x (White, 2015).

La partición de datos se realiza mediante el sistema de archivos distribuido HDFS, con un tamaño de bloque configurado en 128 MB y un factor de replicación de 2, asegurando tolerancia básica a fallos. Cada nodo almacena fragmentos de los conjuntos de datos utilizados en prácticas académicas en tal caso, bases públicas de salud, educación o transporte, permitiendo a los estudiantes experimentar con técnicas de procesamiento paralelo sin requerir acceso a infraestructuras en la nube.

##### **Servicios de software integrados**

El stack de software del LAD incluye:

- Apache Spark, para procesamiento en memoria y tareas de aprendizaje automático.
- Apache Hive, que permite consultas SQL sobre datos almacenados en HDFS, facilitando la transición desde bases de datos tradicionales.
- Apache Pig, para transformaciones de alto nivel en flujos de datos.
- Power BI Desktop (versión gratuita para educación), utilizado en módulos de visualización interactiva.
- Jupyter Notebook y RStudio, como entornos interactivos para programación en Python y R.

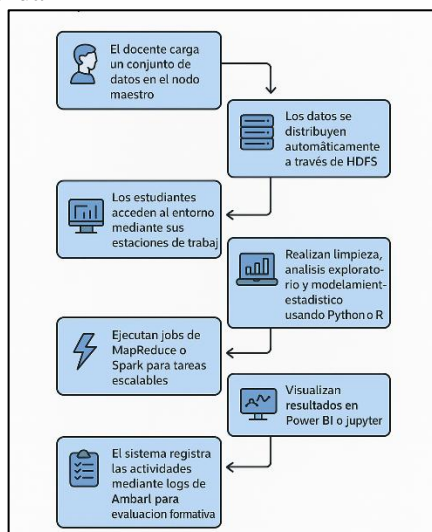
Todos estos servicios son gestionados centralmente mediante Apache Ambari 2.7.7, instalado en el nodo maestro. Ambari proporciona una interfaz web para la monitorización del estado del cluster, la gestión de usuarios, la activación/desactivación de servicios y la generación de alertas ante eventos críticos (Hassin Alasadi & Nemer, 2017). Los agentes de Ambari se instalan en cada estación de trabajo, permitiendo una visibilidad completa del entorno desde una única consola.



### Flujo de trabajo esperado en escenarios académicos

Se ha definido un flujo de trabajo estándar como se muestra en la Figura 4 para las prácticas del LAD:

Figura 4: flujo de trabajo estándar



Este flujo busca integrar competencias técnicas, analíticas y colaborativas en un entorno controlado, alineado con los objetivos de formación en Matemáticas Aplicadas (Márquez Silva & López Martínez<sup>2</sup>, 2025).

### Discusión

El diseño propuesto del Laboratorio Universitario de Análisis de Datos (LAD) responde a una necesidad creciente en los programas de Matemáticas Aplicadas: integrar competencias en ciencia de datos sin depender de infraestructuras externas o servicios en la nube de costo elevado. Este enfoque local y autocontenido es coherente con la tendencia de promover la enseñanza de herramientas de análisis basadas en software de código abierto, especialmente en contextos con recursos limitados (Zhang, Cheng, & Boutaba, 2010).

Una ventaja clave del diseño es su equilibrio entre fidelidad técnica y accesibilidad pedagógica. Al emular un cluster Hadoop mediante modo pseudo-distribuido, se expone a los estudiantes a conceptos fundamentales del cómputo distribuido como partición de datos, replicación y tolerancia a fallos sin requerir hardware especializado ni conocimientos avanzados de redes. Esta estrategia se alinea con lo planteado por White (2015), quien destaca que la comprensión de HDFS y MapReduce no requiere clusters físicos masivos, sino entornos controlados donde los estudiantes puedan observar el comportamiento de los componentes bajo cargas razonables.

La elección de Apache Ambari como capa de gestión no solo simplifica la administración docente, sino que introduce a los estudiantes en prácticas profesionales de monitoreo y operación de infraestructuras de datos. Aunque Ambari no ha sido objeto de estudios empíricos específicos en educación, su adopción en entornos académicos y empresariales se fundamenta en su capacidad para reducir la complejidad operativa de Hadoop, lo cual ha sido ampliamente documentado en su propia documentación técnica y en guías de implementación de código abierto (Apache Software Foundation, 2020).

El uso exclusivo de software de código abierto Python, R, Hadoop, Spark garantiza reproducibilidad, sostenibilidad y accesibilidad, principios esenciales en la educación científica moderna. Como señala McKinney (2010), la estandarización en torno a ecosistemas de código abierto ha transformado la enseñanza de la estadística computacional, permitiendo que los estudiantes construyan portafolios transferibles al mercado laboral. De igual forma, Pedregosa et al. (2011), demuestran cómo bibliotecas como scikit-learn han democratizado el acceso al aprendizaje automático en entornos educativos, incluso sin experiencia previa en programación. La arquitectura propuesta, con ocho estaciones de trabajo y un nodo maestro, está diseñada para manejar conjuntos de datos típicos en proyectos de pregrado generalmente entre 1 y 10 GB, lo

cual es consistente con la práctica real en cursos universitarios de análisis de datos (White, 2015). Aunque no escala a volúmenes de Big Data industrial, su propósito no es replicar centros de datos empresariales, sino formar en los fundamentos del análisis cuantitativo, un objetivo que no requiere infraestructura masiva (O'Neil & Schutt, 2013).

Finalmente, el diseño se diferencia de laboratorios tradicionales al centrarse en el ciclo completo del análisis de datos: desde la adquisición y limpieza hasta la modelización y visualización. Este enfoque integral es coherente con las recomendaciones actuales para la formación en ciencia de datos en carreras cuantitativas, donde la capacidad de interpretar resultados y comunicarlos visualmente es tan importante como la implementación técnica (Wickham, 2016).

### **Conclusiones**

Se propuso el diseño de un Laboratorio Universitario de Análisis de Datos orientado a fortalecer la formación en modelamiento matemático y estadística aplicada dentro de programas de Matemáticas Aplicadas. La arquitectura planteada contempló una infraestructura local compuesta por ocho estaciones de trabajo y un nodo maestro, interconectadas mediante un switch gestionable, configuradas para simular un entorno de cómputo distribuido usando herramientas de código abierto como Hadoop, Python y R. Este enfoque permitió estructurar un modelo técnico factible, centrado en la accesibilidad, sin dependencia de servicios en la nube ni licencias comerciales, y pensado para instituciones con recursos limitados.

El diseño incorporó componentes tanto técnicos como pedagógicos, con el objetivo de integrar la teoría matemática con la práctica analítica. Se definió un flujo de trabajo secuencial desde la carga de datos hasta la visualización de resultados que podría guiar futuras actividades académicas, promoviendo competencias en limpieza, análisis exploratorio, modelamiento y comunicación de hallazgos. Aunque el modelo no ha sido implementado ni evaluado, su estructura responde a principios de simplicidad, reproducibilidad y escalabilidad progresiva, lo que lo convierte en una base viable para futuros desarrollos en contextos universitarios. En conjunto, esta propuesta constituye un marco inicial para la creación de espacios formativos especializados en ciencia de datos, alineados con las demandas actuales del entorno científico y social.

### **Referencias Bibliográficas**

1. Apache Software Foundation. (2020). Apache Ambari Documentation. Obtenido de <https://ambari.apache.org/>
2. Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3, 77–101. Obtenido de <https://doi.org/10.1191/1478088706qp063oa>
3. Camacho Marín, R., Rivas Vallejo, C., Gaspar Castro, M., & Quiñonez Mendoza, C. (2020). Innovación y tecnología educativa en el contexto actual latinoamericano. *Revista de Ciencias Sociales*, 26, 460-472. Obtenido de <https://www.redalyc.org/journal/280/28064146030/html/>
4. CISCO. (2023). Cisco SG200-18 Smart Switch Data Sheet. Cisco. Obtenido de [https://www.cisco.com/c/es\\_mx/obsolete/switches/cisco-small-business-200-series-smart-switches.html](https://www.cisco.com/c/es_mx/obsolete/switches/cisco-small-business-200-series-smart-switches.html)
5. Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, 51, 107–113. Obtenido de <https://doi.org/10.1145/1327452.1327492>
6. Hassin Alasadi, A. H., & Nemer, Z. N. (2017). Finger Vein Verification System based on Three Methodologies of Feature Extraction. *International Journal of Computer Applications*, 172(5), 0975 – 8887. doi:<https://doi.org/10.5120/ijca2017915144>
7. Kolokolov, A., & Zelensky, M. (2024). *Data Visualization with Microsoft Power BI: How to Design Savvy Dashboards*. Sebastopol, California: O'Reilly Media.



8. Márquez Silva, F., & López Martínez<sup>2</sup>, R. (2025). Competencias investigativas y su análisis en el campo de la tecnología educativa mediante e-learning. *Revista Ensayos Pedagógicos*, 20(1), 1-37. doi:<http://doi.org/10.15359/rep.20-1.7>
9. McKinney, W. (2010). Data structures for statistical computing in Python. *Proceedings of the 9th Python in Science Conference*, (págs. 51–56). Obtenido de <https://doi.org/10.25080/Majora-92bf1922-00a>
10. O’Neil, C., & Schutt, R. (2013). *Doing data science: Straight talk from the frontline*. O’Reilly Media.
11. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., & Thirion, B. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2825–2830. Obtenido de <https://doi.org/10.48550/arXiv.1201.0490>
12. Pinto Ayala, B. E., Castañeda Fuentes, J. G., & Sojos Tubay, A. M. (2024). Competencias digitales en docentes latinoamericanos de educación primaria en los años del 2018-2022. *Revista de Ciencias Humanísticas y Sociales*, 49-59. doi:<https://doi.org/10.33936/rehuso.v9i1.5773>
13. Sumbaly, R., Kreps, J., & Wu, L. (2012). The “big data” ecosystem at LinkedIn. *ACM SIGMOD International Conference on Management of Data*, (págs. 1125–1128). Obtenido de <https://doi.org/10.1145/2213836.2213957>
14. Tanenbaum, A., & Wetherall, D. (2011). *Computer networks (5th ed.)*. Pearson Education.
15. White, T. (2015). *Hadoop: The definitive guide (4th ed.)*. O’Reilly Media.
16. Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis (2nd ed.)*. Springer. Obtenido de <https://doi.org/10.1007/978-3-319-24277-4>
17. Yin, R. (2014). *Case study research: Design and methods (5th ed.)*. SAGE Publications.
18. Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1, 7–18. Obtenido de <https://doi.org/10.1007/s13174-010-0007-6>

