UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI

FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

Tema: "Infraestructura para el procesamiento y análisis en algoritmos dirigido para Machine Learning"

Trabajo de Integración Curricular previo a la obtención del título de Ingeniero en Ciencias de la Computación

AUTOR: Borja González Justin Snayder.
TUTOR: Ing. Hidalgo Guijarro Jairo Vladimir. MSc

Tulcán, 2025.

CERTIFICADO DEL TUTOR

Certifico que el estudiante Borja González Justin Snayder con el número de cédula 0924548076 respectivamente ha desarrollado el Trabajo de Integración Curricular: "Infraestructura para el procesamiento y análisis en algoritmos dirigido para Machine Learning"

Este trabajo se sujeta a las normas y metodología dispuesta en el Reglamento de la Unidad de Integración Curricular, Titulación e Incorporación de la UPEC, por lo tanto, autorizo la presentación de la sustentación para la calificación respectiva

Firmado electrónicamente por:
JAIRO VLADIMIR HIDALGO GUIJARRO
Validar únicamente con FirmaEC

_____

Ing. Hidalgo Guijarro Jairo Vladimir. MSc

TUTOR

Tulcán, octubre de 2025

AUTORÍA DE TRABAJO

El presente Trabajo de Integración Curricular constituye un requisito previo para la obtención del título de Ingeniero en la Carrera de ingeniería en informática de la Facultad de Industrias Agropecuarias y Ciencias Ambientales

Yo, Borja González Justin Snayder con cédula de identidad número 0924548076 respectivamente declaro que la investigación es absolutamente original, auténtica, personal y los resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

_____

Borja González Justin Snayder

AUTOR

Tulcán, octubre de 2025

ACTA DE CESIÓN DE DERECHOS DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Yo Borja González Justin Snayder declaro ser autor de los criterios emitidos en el Trabajo de Integración Curricular: "Infraestructura para el procesamiento y análisis en algoritmos dirigido para Machine Learning" y eximo expresamente a la Universidad Politécnica Estatal del Carchi y a sus representantes de posibles reclamos o acciones legales.

_____

Borja González Justin Snayder

AUTOR

Tulcán, octubre de 2025

AGRADECIMIENTO

Quiero expresar mi más profundo agradecimiento a la Universidad Politécnica Estatal del Carchi por ser el pilar de mi formación profesional y personal, por brindarme un espacio de aprendizaje y crecimiento junto a grandes docentes y amigos que marcaron mi camino, a mi tutor, Ing. Jairo Vladimir Hidalgo Guijarro, MSC., por su orientación, dedicación y valioso acompañamiento durante el desarrollo de esta investigación; y a mis amigos, por su apoyo constante, su motivación y por haber hecho de esta etapa una experiencia inolvidable llena de esfuerzo, compañerismo y gratitud.

DEDICATORIA

Dedico este logro a mi madre Miriam González y a mi padre Agustín Borja, quienes han sido mi mayor inspiración en todo que gracias a su amor incondicional y sacrificio pudieron sacarme adelante y a nunca rendirme ellos son mi fortaleza y por siempre creer en mi incluso cuando yo dudaba, gracias por siempre darme su apoyo el cual siempre ha sido una muy importante para mí ustedes siempre serán mi mayor inspiración y modelo para seguir

También a mi querida hermana y sobrina gracias por siempre poder tener su apoyo y compañía en todo este proceso y sé que a pesar de todo siempre pude y siempre voy a poder contar con ustedes

De todo corazón gracias por siempre creer en mí y apoyarme en todo han sido una parte muy importante y fundamental para mí.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

RESUMEN

El presente Trabajo de Integración Curricular diseñado abordo el diseño e implementación de una infraestructura tecnológica distribuida simulada, enfocada al procesamiento de algoritmos de Machine Learning en entornos universitarios. El principal objetivo fue construir un entorno funcional y replicable que mejore el rendimiento y la eficiencia en el entrenamiento de modelos de aprendizaje automático. La metodología utilizada fue de tipo mixta, mediante revisión documental y encuestas a estudiantes, lo cual permitió diagnosticar el estado actual de los laboratorios académicos y validar la pertinencia de la propuesta. Esta implementación se realizó en el único nodo maestro que tiene un sistema operativo CentOS 9, y se le empleo Apache Hadoop 3.3.6, que es un software que divide los datos en bloques de 128 MB y los replica para garantizar tolerancia a fallos, permite escalabilidad horizontal y utiliza HDFS como sistema de almacenamiento distribuido junto con YARN como gestor de recursos. Estuvo apoyado en el modelo MapReduce, el cual separa las tareas en fases de mapeo y reducción, optimizando el análisis de datos en paralelo. también se integró Apache Spark 3.4.1 con PySpark, cuyo motor de ejecución en memoria reduce significativamente los tiempos de respuesta. En este trabajo se implementó ejemplos prácticos con MLlib, como el conteo de palabras en un archivo de 100 caracteres y ejercicios básicos de clasificación, esto permitió validar el funcionamiento del procesamiento distribuido. Cabe destacar que Spark también ofrece otras librerías como Spark SQL para consultas estructuradas y Spark Streaming para procesamiento en tiempo real, respaldadas por un planificador basado en DAG (Directed Acyclic Graph) que optimiza la ejecución de tareas distribuidas. Se determino que este modelo es viable, económico y adaptable para universidades que buscan fortalecer sus capacidades investigativas en ciencia de datos e inteligencia artificial, impulsando la transformación digital en la educación superior.

Palabras clave: Machine Learning, Computación Distribuida , Apache Hadoop, Apache Spark, PySpark.

ABSTRACT

This Curricular Integration Work focused on the design and implementation of a simulated distributed technological infrastructure aimed at processing Machine Learning algorithms in university environments. The research applied a mixed methodology, combining documentary review and student surveys, to assess the current state of academic laboratories and validate the feasibility of the proposal. The implementation was carried out on a single master node with CentOS 9, using Apache Hadoop 3.3.6, which divides data into 128 MB blocks, replicates them to ensure fault tolerance, allows horizontal scalability, and manages resources through HDFS and YARN. The processing relied on the MapReduce model for parallel analysis, while Apache Spark 3.4.1 with PySpark was integrated to take advantage of its in-memory execution engine. Practical examples were executed with MLlib, including a word count on a 100-character file and basic classification exercises, confirming the operation of distributed processing. It is also highlighted that Spark provides other libraries such as Spark SQL for structured queries and Spark Streaming for real-time processing, supported by a DAG-based scheduler that optimizes distributed task execution. The results demonstrated that this model is viable, cost-effective, and adaptable for universities seeking to strengthen their research capabilities in data science and artificial intelligence, contributing to digital transformation in higher education.

Keywords: Machine Learning, Distributed Computing, Apache Hadoop, Apache Spark, PySpark.

INTRODUCCIÓN

Hoy en día, las universidades enfrentan un proceso acelerado de transformación digital, impulsado por la necesidad de gestionar grandes volúmenes de información y realizar análisis avanzados en diversos campos del conocimiento. Dentro de este contexto, el aprendizaje automático (machine learning) se ha consolidado como una herramienta esencial para la innovación tecnológica, la investigación de vanguardia y la generación de soluciones en áreas como la salud, las ciencias sociales, las ingenierías y la educación.

Sin embargo, muchas instituciones de educación superior aún carecen de infraestructuras tecnológicas capaces de sostener la ejecución eficiente de algoritmos complejos. El uso de infraestructuras tradicionales o genéricas limita la capacidad de entrenar modelos predictivos, analizar grandes conjuntos de datos y aprovechar al máximo el poder de cómputo que demandan los algoritmos actuales. Esta carencia restringe la innovación científica y afecta la competitividad de las universidades en la producción de nuevo conocimiento (García et al., 2022).

En este escenario, los sistemas distribuidos surgen como una alternativa estratégica. Un sistema distribuido se define como un conjunto de computadoras independientes, interconectadas mediante una red, que funcionan de manera coordinada y se presentan a los usuarios como un único sistema coherente (Tanenbaum & Van Steen, 2017). Este tipo de arquitectura ofrece ventajas como la escalabilidad, la tolerancia a fallos y la posibilidad de procesar datos de forma paralela, lo cual resulta fundamental para entornos académicos que requieren eficiencia y flexibilidad en la investigación con machine learning.

De acuerdo con la UNESCO (2023), las instituciones educativas necesitan adoptar modelos tecnológicos que aseguren un acceso equitativo a recursos de cómputo

avanzados, fomentando la investigación y la formación en inteligencia artificial. En este sentido, la implementación de centros de cómputo distribuidos o clústeres universitarios permite optimizar la ejecución de algoritmos, gestionar grandes volúmenes de información y potenciar la colaboración interdisciplinaria entre estudiantes e investigadores.

La presente tesis se centra en el diseño y desarrollo de una infraestructura tecnológica distribuida orientada al procesamiento de algoritmos de machine learning en universidades. Al atender las demandas específicas del ámbito académico, esta propuesta busca no solo mejorar la eficiencia investigativa, sino también fortalecer la formación de competencias digitales en docentes y estudiantes. Asimismo, se plantea un modelo replicable para otras instituciones, promoviendo prácticas tecnológicas sostenibles y contribuyendo al avance de la educación superior en la era digital.

I. EL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

En la era de la digitalización, las universidades enfrentan un crecimiento exponencial en la generación de datos derivados de diversas fuentes: investigaciones científicas, simulaciones en tiempo real, sistemas automatizados y herramientas de enseñanza-aprendizaje. Este fenómeno es particularmente notable en los laboratorios de cómputo universitarios, donde la aplicación práctica de algoritmos, modelos matemáticos y soluciones basadas en inteligencia artificial requiere de entornos altamente demandantes en procesamiento de datos y capacidad computacional. No obstante, muchos de estos laboratorios aún operan con infraestructuras tradicionales, poco escalables o desactualizadas, lo que impide el aprovechamiento eficiente de los grandes volúmenes de datos generados.

La problemática central radica en la ausencia de infraestructuras especializadas y sistemas adecuados para el manejo de Big Data en entornos universitarios, especialmente en los laboratorios de cómputo, lo cual limita considerablemente las oportunidades de formación práctica en áreas emergentes como el aprendizaje automático (Machine Learning), la ciencia de datos y la inteligencia artificial. Esta situación afecta tanto a docentes como a estudiantes e investigadores, quienes encuentran barreras técnicas para implementar soluciones innovadoras, experimentar con modelos de análisis predictivo o ejecutar proyectos que requieren procesamiento masivo y simultáneo de datos. Según (González et al., 2023), la falta de entornos tecnológicos adecuados en instituciones educativas compromete la capacidad de innovar, de tomar decisiones basadas en evidencia y de preparar profesionales que respondan a los desafíos del mercado laboral digital actual.

ejecuciones compatibles laboratorios universitarios no cuentan con plataformas que integren herramientas de análisis distribuidos como Hadoop, Spark o entornos de ejecución compatibles con Tensor Flow, PyTorch o Scikit-learn. Esto impide el desarrollo

de proyectos que requieren la paralelización de procesos, la gestión de grandes datasets o la experimentación con redes neuronales complejas. Como consecuencia, la formación académica se ve limitada a enfoques teóricos, sin el complemento práctico necesario para adquirir competencias reales en la implementación de modelos de aprendizaje automático. Esta brecha entre teoría y práctica reduce significativamente la competitividad de los egresados y afecta negativamente la calidad de la educación superior.

Por otro lado, la ejecución de investigaciones científicas orientadas a la solución de problemas reales en diversos sectores productivos también se ve comprometida. La capacidad de procesar datos en tiempo real, de aplicar algoritmos de clasificación, predicción y detección de patrones, depende directamente de la infraestructura tecnológica con la que cuenta el laboratorio universitario. Ramírez y Pérez (2022), sostienen que el aprendizaje automático, cuando es respaldado por una infraestructura robusta, permite automatizar procesos complejos, realizar análisis en profundidad y obtener hallazgos claves y estratégicos que impactan directamente en la toma de decisiones. Sin embargo, este potencial no puede ser explotado si no existen las condiciones técnicas mínimas para su aplicación.

En el contexto ecuatoriano y latinoamericano, muchas instituciones de educación superior enfrentan desafíos relacionados con la obsolescencia tecnológica de sus laboratorios, la falta de inversión en infraestructura y la ausencia de políticas institucionales que prioricen la innovación digital. Esta situación se traduce en una baja eficiencia investigativa, en una limitada participación en redes internacionales de ciencia de datos, y en la escasa producción de soluciones tecnológicas basadas en algoritmos. De acuerdo con López et al. (2021), el establecimiento de centros de cómputo especializados y dotados con tecnologías de punta representa una vía efectiva para mejorar la calidad de la investigación, promover el desarrollo de prototipos funcionales y fortalecer el vínculo entre academia e industria.

Además, la creciente demanda por profesionales capacitados en el uso de herramientas de análisis masivo de datos, tanto en el sector público como privado, exige a las universidades modernizar sus infraestructuras de laboratorio. No se trata únicamente de incorporar hardware de alto rendimiento, sino de diseñar entornos integrales que permitan simular, procesar, analizar y visualizar datos en tiempo real. La implementación de estos entornos no solo beneficiará la calidad académica, sino que también permitirá el desarrollo de proyectos interdisciplinares, colaborativos e

innovadores. La infraestructura, entonces, se convierte en un eje transversal para el aprendizaje significativo y la formación de competencias digitales avanzadas.

Por todo lo anterior, se evidencia la necesidad de desarrollar una infraestructura eficiente para el procesamiento y análisis de datos algorítmicos, dirigida especialmente a laboratorios universitarios de cómputo. Este tipo de infraestructura debe ser escalable, flexible, modular y adaptable a diferentes contextos educativos, permitiendo la integración de tecnologías como el aprendizaje automático y el procesamiento distribuido. La presente investigación tiene como propósito diseñar y validar un prototipo de infraestructura tecnológica orientada a laboratorios académicos, que permita potenciar la capacidad analítica, investigativa y formativa en el contexto de la educación superior, promoviendo así una transformación digital sostenible y alineada a los retos del siglo XXI.

1.2. FORMULACIÓN DEL PROBLEMA

¿Cómo puede un sistema distribuido contribuir en el procesamiento de grandes volúmenes de datos y en el análisis en algoritmos para machine learning y en la mejora de la toma de decisiones y la gestión eficiente de la investigación en una universidad?

1.3. JUSTIFICACIÓN

Desde este contexto educativo actual se está caracterizando por el crecimiento de datos por lo cual es imprescindible asegurar una infraestructura sólida para el manejo y análisis de grandes cantidades de datos en, donde la habilidad para examinar información de forma eficaz se ha transformado en un elemento crucial para optimizar la administración académica, administrativa y de investigación.

La implementación de infraestructuras especializadas para el tratamiento de algoritmos de aprendizaje automático es una táctica esencial para mejorar el análisis de datos en universidades, facilitando la superación de las restricciones de las herramientas convencionales que no están concebidas para gestionar grandes cantidades de datos y exigencias complejas de análisis predictivo.

Las infraestructuras enfocadas en el procesamiento distribuido y el análisis de datos mediante el aprendizaje automático se han transformado en instrumentos esenciales para las universidades contemporáneas, simplificando la automatización de procesos la predicción de tendencias y la toma de decisiones fundamentadas en

datos, numerosas instituciones educativas continúan apoyándose en sistemas no especializados o instrumentos convencionales, lo que conduce a decisiones ineficientes, restricciones en la personalización de los servicios de educación y ausencia de habilidad para prever retos futuros.

La implementación de estas infraestructuras en instituciones educativas tiene el objetivo de aportar de manera significativa a la mejora de los procesos académicos y administrativos, incrementando la eficacia operacional y la habilidad para ajustarse a las demandas variables de los alumnos y de los ambientes educativos. La automatización de las evaluaciones de datos facilitará una administración más eficaz de los recursos y una toma de decisiones más exacta.

El propósito de este proyecto es responder a la necesidad urgente de modernización tecnológica mediante la implementación de una infraestructura especializada para el manejo de algoritmos de aprendizaje automático. El objetivo es desarrollar una solución integral que simplifique el manejo eficiente de grandes volúmenes de información, desde la gestión académica hasta la investigación, adaptándose a las necesidades específicas de las universidades y fortaleciendo su competitividad en el ámbito académico y científico.

Asimismo, el proyecto aspira a crear un modelo replicable para otras instituciones universitarias que se encuentran con retos parecidos en la administración y el manejo de datos. Al mostrar las ventajas palpables de la aplicación del aprendizaje automático en la educación, se busca impulsar una revolución digital en el ámbito educativo, brindando a las instituciones instrumentos que mejoren sus procesos y posibiliten un incremento sostenible en la calidad de la educación.

1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN

1.4.1. Objetivo General

Diseñar e implementar una infraestructura de computación distribuida de software libre para el procesamiento de algoritmos de Machine Learning.

1.4.2. Objetivos Específicos

- Analizar los principios básicos y las diversas arquitecturas de computación distribuida, enfocándose en su aplicación al procesamiento y análisis de algoritmos de machine learning

- Integrar herramientas de código abierto en un entorno distribuido para facilitar la ejecución eficiente de algoritmos de machine learning
- Desarrollar un prototipo funcional de una infraestructura distribuida que este orientado al procesamiento de algoritmos de machine learning

1.4.3. Preguntas de Investigación

- ¿Qué elementos tecnológicos son cruciales para edificar una infraestructura eficaz para el procesamiento de análisis en algoritmos de aprendizaje automático en una universidad?
- ¿De qué manera influye una infraestructura distribuida para machine learning en el desarrollo y en la optimización de procesos de investigación?
- ¿Cómo una infraestructura específica para el manejo de algoritmos de aprendizaje automático puede impulsar la investigación académica en la universidad?

II. FUNDAMENTACIÓN TEÓRICA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

En el campo del manejo de grandes cantidades de información y la puesta en marcha de infraestructura para algoritmos de aprendizaje automático, numerosas investigaciones han sugerido soluciones revolucionarias, subrayando la relevancia de la eficacia y escalabilidad de los sistemas.

Gómez (2023) creó una arquitectura distribuida destinada a mejorar los modelos de aprendizaje profundo en tiempo real en la Universidad Nacional de Ingeniería. Mediante el uso de Apache Spark y Tensor Flow, el sistema logró disminuir en un 40% los periodos de entrenamiento, potenciando de manera notable la habilidad para procesar datos obtenidos de sensores en tiempo real. El enfoque empleado se fundamentó en métodos híbridos que fusionaban el almacenamiento en la nube y el procesamiento en el borde (Gómez, Arquitecturas distribuidas para aprendizaje profundo, 2023).

Martínez (2022) desarrolló un sistema de administración para el procesamiento en masa de datos en clústeres económicos para usos de inteligencia artificial en Ecuador. Este proyecto puso en marcha Hadoop y Spark en hardware asequible, consiguiendo una relación ventajosa entre costo y beneficio para pequeñas y medianas compañías de tecnología. Los hallazgos subrayaron la relevancia de establecer arquitecturas versátiles y adaptativas para optimizar el desempeño computacional (Martinez, Procesamiento de datos en clústeres económicos: Un enfoque para inteligencia artificial, 2022).

Ruiz (2021) mostró un prototipo de infraestructura escalable destinado a instruir algoritmos de aprendizaje automático en grandes volúmenes de datos climáticos. La investigación se enfocó en la creación de un sistema que combinaba recursos GPU y ambientes Docker para la implementación eficaz de modelos. En consecuencia, se consiguió un incremento del 50% en el desempeño informática, lo que simplifica la predicción de fenómenos climáticos extremos. (Ruíz, 2021)

Lee y colaboradores (2020) analizaron el efecto de la implementación de FPGAs (Arrays de Gate Programable Field) en la mejora del procesamiento de algoritmos de inteligencia artificial en Corea del Sur. El estudio determinó que estas tecnologías tienen el potencial de incrementar la rapidez de procesamiento en un 70% en comparación con sistemas tradicionales que se fundamentan en CPU y GPU, en particular en aplicaciones vitales como el reconocimiento de imágenes y la identificación de fraudes (Lee et al., 2020)

2.2. MARCO TEÓRICO

2.2.1 Automatización en Infraestructuras Universitarias para Machine Learning

La automatización en entornos universitarios facilita el diseño, entrenamiento y despliegue de modelos de aprendizaje automático, reduciendo el tiempo y los errores humanos en tareas repetitivas. Según (Fernández y Ríos, 2023), automatizar procesos en el ámbito educativo mejora la eficiencia operativa y optimiza la asignación de recursos tecnológicos en proyectos de ciencia de datos. Esta automatización es clave para gestionar datos masivos, procesos de entrenamiento distribuido y monitoreo de modelos en tiempo real dentro de laboratorios académicos.

2.2.2 Computación Distribuida en Ambientes Académicos

La computación distribuida es esencial en el desarrollo de soluciones de Machine Learning a gran escala. Permite la ejecución paralela de tareas en distintos nodos, acelerando el procesamiento de datos y la eficiencia en la formación de modelos. En contextos universitarios, tecnologías como Apache Hadoop y Spark son ampliamente usadas para implementar infraestructuras capaces de manejar volúmenes de datos crecientes (Matei et al., 2023). Estas plataformas permiten a estudiantes e investigadores realizar simulaciones, análisis predictivos y procesamiento de datos en tiempo real.

2.2.3 Frameworks Avanzados para el Desarrollo de Algoritmos: TensorFlow y PyTorch

TensorFlow y PyTorch son las bibliotecas más utilizadas en la investigación académica y el desarrollo profesional de modelos de aprendizaje automático. TensorFlow destaca por su robustez y escalabilidad para la producción, mientras que PyTorch es preferido por su flexibilidad y dinamismo en entornos de investigación (Zhao y Liu, 2023). Ambos frameworks son compatibles con plataformas distribuidas y

aceleradores como GPUs y TPUs, lo que los convierte en herramientas indispensables en laboratorios de cómputo universitarios.

2.2.4 Herramientas de Orquestación, Procesamiento y Virtualización en Infraestructuras Académicas

El uso de herramientas modernas de virtualización y procesamiento distribuido es fundamental para soportar la ejecución eficiente de modelos de Machine Learning. Algunas de las más relevantes incluyen:

- Docker y Kubernetes: permiten empaquetar, desplegar y escalar aplicaciones ML en contenedores distribuidos (CNCF, 2023).
- Apache Hadoop: ideal para almacenamiento distribuido y procesamiento masivo de datos (White, 2022).
- Apache Spark: permite procesamiento en memoria, mejorando el rendimiento de tareas de aprendizaje (Matei et al., 2023)
- Dask y Ray: bibliotecas que permiten escalar procesos de análisis y entrenamiento de modelos en Python (Rocklin, 2023).
- Apache Airflow: útil para automatizar flujos de trabajo de ciencia de datos (ASF, 2023).
- MLflow: facilita la gestión del ciclo de vida de los modelos, seguimiento de experimentos y despliegue (Zaharia, 2020).
- JupyterHub: permite a múltiples usuarios acceder a entornos de experimentación colaborativa, crucial en educación superior (Project Jupyter, 2023).

2.2.5 Escalabilidad y Rendimiento en Sistemas de Inteligencia Artificial

La escalabilidad es un requisito indispensable en sistemas académicos de inteligencia artificial. La posibilidad de ajustar la infraestructura al volumen de datos o usuarios permite mantener un rendimiento adecuado. PwC (2023) destaca que una arquitectura escalable no solo mejora la velocidad de entrenamiento, sino que también permite a las instituciones expandir sus capacidades computacionales según las necesidades del proyecto sin sacrificar precisión.

2.2.6 Infraestructura como Servicio (IaaS) y su Aplicación Académica

La Infraestructura como Servicio (IaaS) permite a las universidades acceder a recursos de cómputo avanzados sin necesidad de inversión física en servidores. Servicios como

OpenStack, implementados en nubes académicas, facilitan la creación de clústeres virtuales para pruebas, prototipos y entrenamiento de modelos. IBM (2023) señala que IaaS es clave en educación para fomentar la equidad en el acceso a tecnologías avanzadas.

2.2.7 Cuadro comparativo de arquitecturas para Machine learning

Tabla 1. Evaluación comparativa de arquitecturas de Machine Learning

Comparativo de Arquitecturas para Aprendizaje Automático

Arquitectura / Herramienta	Características Principales	Ventajas	Desventajas	Casos de Uso Común
Apache Hadoop	Sistema de almacenamiento distribuido (HDFS) + procesamiento por lotes (MapReduce)	Escalable, tolerante a fallos	Menor velocidad para tareas interactivas	Procesamiento batch, almacenamiento masivo
Apache Spark	Procesamiento en memoria, incluye MLlib para ML	Alta velocidad, integración con Python (PySpark)	Mayor consumo de memoria	Análisis de datos, entrenamiento de modelos
Docker + Kubernetes	Contenedores para aplicaciones + orquestación	Flexible, reproducible, escalable	Requiere conocimientos técnicos	Laboratorios de investigación, ambientes de prueba
Ray	Librería de Python para tareas distribuidas	Fácil de usar, ideal para ML y RL	Menos maduro que Spark	Entrenamiento paralelo, tareas de IA
Dask	Ejecuta operaciones tipo Pandas/Numpy en paralelo	Ligero, fácil integración con Jupyter	Escalabilidad limitada	Educación práctica, análisis básico de datos
TensorFlow Distributed	Entrenamiento distribuido en GPU/TPU	Optimizado para Deep Learning, integración con GCP	Complejo de configurar	Entrenamiento de redes neuronales profundas
OpenStack (IaaS)	Plataforma para nubes privadas académicas	Escalabilidad, control total de recursos	Requiere hardware y soporte dedicado	Infraestructura virtualizada para universidades

2.2.8 Ética y Privacidad en el Manejo de Datos de Machine Learning

En contextos educativos, el uso de datos requiere una atención especial a la ética y la privacidad. La (UNESCO, 2023) establece que los proyectos de inteligencia artificial deben cumplir principios de transparencia, explicabilidad y protección de datos

sensibles. Para la implementación de infraestructuras tecnológicas en universidades, estos principios se traducen en la necesidad de aplicar políticas institucionales claras sobre el tratamiento de la información de los estudiantes y docentes.

2.2.9 Tendencias Tecnológicas 2023–2025 en Infraestructura para Machine Learning

Entre las principales tendencias para los próximos años están el uso de modelos multimodales, agentes inteligentes colaborativos, la implementación de inteligencia artificial explicativa y el avance de la computación cuántica. Según (TechTarget, 2023), estos cambios demandan infraestructuras flexibles, escalables y altamente automatizadas, especialmente en contextos universitarios donde se debe experimentar e innovar constantemente.

2.2.10 Modelos Replicables de Infraestructura Educativa para Machine Learning

Diseñar infraestructuras que puedan replicarse en distintas universidades es esencial para fomentar la investigación descentralizada. Estudios como el de (González y Ramírez, 2023) proponen arquitecturas modulares que integran contenedores, redes definidas por software (SDN) y plataformas de automatización para ofrecer soluciones sostenibles y adaptables a cualquier institución educativa.

2.2.11 Estrategias Metodológicas en el Desarrollo de Software

El proceso de desarrollo de software moderno exige el uso de metodologías que orienten la planificación, diseño, implementación y validación de los sistemas de manera eficiente y estructurada. La selección de una metodología adecuada permite mejorar la productividad de los equipos, adaptarse a entornos cambiantes y reducir los riesgos asociados al proyecto (Pressman y Maxim, 2021)

Metodologías Ágiles

Las metodologías ágiles son ampliamente utilizadas por su capacidad de adaptación frente a cambios en los requisitos y su enfoque en la entrega continua de valor. Este enfoque propone ciclos iterativos cortos que permiten retroalimentación frecuente y validación continua del producto. Según (Serrador y Pinto, 2025), factores como la colaboración efectiva, la comunicación constante y la participación del cliente son esenciales para mejorar la productividad del equipo de desarrollo.

Scrum

Scrum es un marco de trabajo ágil basado en entregas incrementales a través de iteraciones llamadas *sprints*. De acuerdo con (Schwaber y Sutherland, 2020), Scrum facilita la transparencia, la inspección y la adaptación continua del proceso. La implementación de roles como *Scrum Master*, *Product Owner* y *Development Team* garantiza la gestión estructurada de las tareas y la toma rápida de decisiones.

Extreme Programming (XP)

XP es una metodología ágil centrada en la mejora continua, la simplicidad del diseño y la obtención temprana de retroalimentación. Iqbal et al. (2022) destacan que XP incluye prácticas como desarrollo guiado por pruebas, integración continua y programación en parejas, las cuales aumentan la calidad del software y promueven entornos colaborativos.

Metodologías Tradicionales

Los modelos tradicionales, como el modelo en cascada, estructuran el proceso de desarrollo en fases secuenciales y rígidas. Estos enfoques son adecuados para proyectos con requisitos bien definidos desde el inicio, pero limitan la flexibilidad para gestionar cambios. Según (Rodríguez y Sánchez, 2022), estos modelos permiten una mayor trazabilidad y documentación del proceso, siendo comunes en sistemas embebidos y proyectos de gran escala.

Modelo en Cascada

Este modelo define etapas bien delimitadas como análisis, diseño, implementación, verificación y mantenimiento. Cada fase se completa completamente antes de pasar a la siguiente. (Sommerville, 2020) señala que este enfoque puede volverse ineficiente ante la necesidad de retrocesos o cambios posteriores, lo cual lo vuelve poco ideal para entornos dinámicos.

Modelo en Espiral

El modelo en espiral combina elementos estructurados e iterativos. (Blašković y Čandrlić, 2021)explican que este modelo prioriza la gestión del riesgo mediante iteraciones cíclicas en las que se identifican, evalúan y mitigan posibles problemas antes de avanzar. Es adecuado para proyectos complejos que requieren alta adaptabilidad y análisis continuo.

Tabla 2. Cuadro comparativo en estrategias Metodológicas en el Desarrollo de Software

Comparación de Estrategias para Proyectos de Software		
Metodología	**Ventajas**	**Desventajas**
Scrum	• Permite ver resultados en poco tiempo • Mejora la comunicación entre el equipo • Es fácil adaptarse si algo cambia	• No funciona bien si el equipo no se organiza • Requiere compromiso de todos • Puede ser difícil en grupos muy grandes
XP (Programación Extrema)	• Mejora la calidad del código • Se adapta rápido a los cambios • Favorece el trabajo en equipo	• Necesita mucha disciplina • No es buena opción para proyectos con poca interacción • Puede no ser entendida fácilmente por todos
Cascada	• Es fácil de seguir paso a paso • Deja todo bien documentado • Útil si el proyecto es muy claro desde el inicio	• Es complicado hacer cambios una vez que se avanza • Puede tardar mucho en mostrar resultados • No permite mucha flexibilidad
Espiral	• Permite detectar problemas a tiempo • Combina lo mejor de varios modelos • Ideal para proyectos grandes con riesgos	• Es más costosa de aplicar • Puede volverse compleja si no se gestiona bien • No es buena si el proyecto es muy simple

2.2.11.1 Metodología de Desarrollo: Modelo en Cascada

Para el desarrollo de la infraestructura tecnológica propuesta, se emplea el modelo de desarrollo en cascada, una metodología estructurada y secuencial que permite avanzar por fases bien definidas, asegurando la completitud y calidad de cada etapa antes de pasar a la siguiente. El modelo en cascada resulta apropiado para proyectos donde los requisitos están bien definidos desde el inicio y el desarrollo técnico requiere una planificación detallada, como en el caso de la implementación de sistemas distribuidos de Machine Learning en entornos universitarios (Sommerville, 2020).

2.2.11.2 Fases del modelo en cascada aplicado al proyecto

- **Recolección de Requisitos**: Se recopila información sobre el contexto universitario, estado actual de los laboratorios, y necesidades técnicas para la ejecución de algoritmos de ML.
- **Análisis del Sistema**: Se identifican herramientas, plataformas y recursos necesarios para el diseño de la infraestructura.

- **Diseño del Sistema**: Se plantea una arquitectura modular y escalable que incluye servidores con GPU, contenedores y sistemas distribuidos.
- **Implementación**: Se lleva a cabo la instalación, configuración e integración de las tecnologías seleccionadas.
- **Pruebas**: Se realizan evaluaciones funcionales y de rendimiento en escenarios reales.
- **Mantenimiento**: Se documenta el sistema y se definen protocolos para futuras mejoras y escalabilidad.

2.2.11.3 Justificación del modelo en cascada

El modelo en cascada permite una organización lineal, ideal para proyectos donde las etapas pueden ser abordadas de forma secuencial y los objetivos están claramente definidos desde el inicio. En esta investigación, su uso asegura una construcción sólida y documentada del prototipo, adaptado a las condiciones reales del entorno académico.

2.2.12 Impacto de la Infraestructura Tecnológica en la Investigación Universitaria

La infraestructura tecnológica es un factor clave para potenciar la investigación académica en las universidades. Según (García et al., 2022), la disponibilidad de recursos computacionales avanzados permite reducir el tiempo de procesamiento de modelos, ampliar el alcance de las investigaciones interdisciplinarias y facilitar la experimentación en áreas como inteligencia artificial, bioinformática, análisis climático, entre otras.

En entornos de ciencia de datos, contar con plataformas distribuidas permite a los investigadores ejecutar simulaciones complejas, analizar grandes volúmenes de información y validar modelos predictivos de forma más eficiente. Esto no solo mejora la productividad científica, sino que también aumenta la capacidad de generar soluciones aplicables a problemas reales del entorno.

Además, la infraestructura adecuada favorece el acceso equitativo a herramientas avanzadas, democratizando el conocimiento y permitiendo la inclusión de grupos de investigación con menos recursos. Tal como lo establece la (UNESCO, 2023), el acceso a tecnologías de análisis de datos e inteligencia artificial debe ser una prioridad para cerrar brechas académicas y científicas entre regiones.

2.2.13 Relación entre Computación Distribuida y Producción Científica Basada en Datos

La computación distribuida, al permitir la ejecución paralela de procesos y el manejo eficiente de datos a gran escala, se alinea con las necesidades de los sistemas de investigación modernos. De acuerdo con la (UNESCO, 2023) las universidades que invierten en infraestructura tecnológica avanzada logran fortalecer sus capacidades investigativas, acelerar la producción científica y mejorar la calidad de sus contribuciones académicas. La implementación de plataformas distribuidas facilita el desarrollo de investigaciones colaborativas, el uso compartido de recursos computacionales y la experimentación con modelos complejos, lo cual impacta directamente en la innovación institucional y la competitividad científica.

2.2.14 Apache Hadoop y su Ecosistema de Servicios en Infraestructuras Académicas para Machine Learning

Apache Hadoop es un marco de trabajo de código abierto diseñado para el procesamiento distribuido y fiable de grandes volúmenes de datos en clústeres de hardware convencional. Su arquitectura modular permite escalar desde un solo servidor hasta miles de máquinas, gestionando petabytes de información de manera eficiente y tolerante a fallos.

Características Fundamentales de Hadoop

- Confiabilidad y tolerancia a fallos: Hadoop crea múltiples copias automáticas de los datos. Si ocurre un fallo, el sistema recupera la información y reintegra los nodos afectados al clúster, tratando los fallos de hardware como una situación habitual y no excepcional.
- Escalabilidad: Permite el escalado horizontal, distribuyendo tanto los datos como el procesamiento entre múltiples servidores, lo que lo hace apto para infraestructuras académicas que requieren flexibilidad ante el crecimiento de datos y usuarios.
- Portabilidad: Se puede instalar en diferentes tipos de hardware y sistemas operativos, facilitando su adopción en entornos universitarios con recursos heterogéneos.

- Procesamiento distribuido: Utiliza modelos de programación sencillos (como MapReduce) para dividir y paralelizar tareas sobre grandes conjuntos de datos.

Componentes Principales del Ecosistema Hadoop

- Hadoop Distributed File System (HDFS): Sistema de archivos distribuido que almacena los datos de manera redundante en el clúster, garantizando alta disponibilidad y acceso eficiente incluso ante fallos de nodos.
- MapReduce: Motor de procesamiento paralelo que divide las tareas en subtareas (map) y luego las combina (reduce), optimizando el análisis de datos masivos.
- YARN (Yet Another Resource Negotiator): Gestor de recursos que coordina la ejecución de aplicaciones distribuidas, maximizando la utilización de los recursos del clúster.
- Hadoop Common: Conjunto de utilidades y bibliotecas necesarias para el funcionamiento de los demás módulos.

Servicios y Herramientas del Ecosistema Hadoop

El ecosistema Hadoop se ha expandido para incluir una variedad de servicios y herramientas que potencian su funcionalidad en proyectos de Machine Learning y ciencia de datos:

- Apache Hive: Permite consultas SQL sobre grandes volúmenes de datos almacenados en HDFS, facilitando el análisis para usuarios no expertos en programación.
- Apache Pig: Lenguaje de alto nivel para la transformación y análisis de datos, ideal para flujos de procesamiento complejos.
- Apache HBase: Base de datos NoSQL distribuida que permite el acceso en tiempo real a grandes conjuntos de datos, útil para aplicaciones que requieren baja latencia.
- Apache Spark: Aunque es un proyecto independiente, se integra con Hadoop para aprovechar HDFS y realizar procesamiento en memoria, acelerando tareas iterativas y de Machine Learning.
- Apache ZooKeeper: Servicio de coordinación distribuida que gestiona la configuración y sincronización de los diferentes componentes del clúster.

- Apache Oozie: Orquestador de flujos de trabajo para programar y gestionar tareas complejas dentro del ecosistema Hadoop.

Aplicaciones de Hadoop en el Ámbito Académico

En la educación superior, Hadoop es clave para la gestión y análisis de grandes volúmenes de datos generados por plataformas de aprendizaje, sistemas administrativos y proyectos de investigación en inteligencia artificial y ciencia de datos. Según (Gonzalez et al., 2020), Hadoop y HDFS permiten construir arquitecturas tipo data lake que maximizan la disponibilidad y accesibilidad de datos para aplicaciones analíticas, algo esencial para la investigación educativa moderna.

La literatura reciente resalta que, sin soluciones como Hadoop, sería inviable manejar la creciente cantidad de datos producidos en entornos académicos, tanto para la evaluación educativa como para la innovación en metodologías didácticas basadas en datos (Soto et al., 2023).

Limitaciones y Tendencias Actuales

Aunque Hadoop es robusto para el procesamiento batch, presenta limitaciones en tareas iterativas y de baja latencia, fundamentales en inteligencia artificial. Por ello, muchas instituciones complementan Hadoop con Spark, que mejora significativamente el rendimiento en estos escenarios. Además, la tendencia actual es migrar hacia soluciones cloud que simplifican la instalación, gestión y escalabilidad de clústeres Hadoop, permitiendo a las universidades centrarse en la investigación y el desarrollo de modelos sin preocuparse por la infraestructura física.

2.2.15 Apache Spark y PySpark como Núcleo de Procesamiento Distribuido para Machine Learning

Apache Spark es un motor de procesamiento distribuido de código abierto diseñado para ejecutar tareas complejas sobre grandes volúmenes de datos con alta velocidad y eficiencia. Su arquitectura está basada en el uso de estructuras de datos distribuidas resilientes (RDDs), lo que le permite tolerar fallos y escalar horizontalmente a múltiples nodos sin comprometer el rendimiento (Tang et al., 2020).

Este marco ofrece bibliotecas especializadas como:

- **Spark SQL**, que permite consultas estructuradas con sintaxis similar a SQL;
- **Spark Streaming**, para procesamiento en tiempo real;

- **GraphX**, orientado al análisis de grafos;
- y **MLlib**, un módulo para algoritmos de aprendizaje automático, incluyendo clasificación, regresión, clustering y reducción de dimensionalidad (Meng et al., 2022).

Una de sus mayores fortalezas es el procesamiento en memoria, lo cual reduce significativamente el tiempo de ejecución en comparación con tecnologías como Hadoop MapReduce. Esto lo convierte en una opción ideal para tareas iterativas como entrenamiento de modelos de Machine Learning (Dünner et al., 2021).

Por su parte, PySpark es la interfaz de Apache Spark desarrollada para Python. Permite a investigadores y desarrolladores aplicar el poder de Spark sin necesidad de programar en Scala o Java, facilitando su integración con herramientas ampliamente utilizadas como pandas, NumPy y Jupyter Notebooks.

PySpark también ofrece compatibilidad con frameworks de inteligencia artificial como TensorFlow y PyTorch a través de conectores o integraciones especializadas, y ha incorporado recientemente funciones como los UDTFs y aceleración con Apache Arrow para aumentar el rendimiento (Databricks, 2023).

Gracias a su versatilidad y escalabilidad, Apache Spark y PySpark se consolidan como pilares fundamentales para la infraestructura de datos en entornos académicos, permitiendo a los laboratorios universitarios ejecutar simulaciones, pruebas y análisis avanzados en clústeres distribuidos, incluso con recursos limitados.

III. METODOLOGÍA

3.1. ENFOQUE METODOLÓGICO

3.1.1. Enfoque

Este estudio se desarrolló bajo un enfoque metodológico mixto, el cual combina herramientas cualitativas y cuantitativas para obtener una visión más integral del problema y de la solución planteada. Esta combinación metodológica permite abordar tanto los aspectos técnicos como humanos que rodean la implementación de una infraestructura distribuida para el procesamiento de algoritmos de Machine Learning en el entorno universitario.

3.1.1.1 Enfoque Cualitativo

En este estudio, el componente cualitativo se desarrolló mediante una revisión documental y análisis bibliográfico de literatura científica, artículos académicos y reportes institucionales relacionados con el uso de infraestructuras tecnológicas distribuidas en entornos educativos. Esta revisión permitió identificar tendencias actuales, limitaciones presentes en universidades y experiencias previas de implementación de sistemas distribuidos aplicados al aprendizaje automático. Gracias a este análisis se obtuvo una visión integral sobre la viabilidad de adoptar una infraestructura de este tipo, apoyando el desarrollo de procesos de investigación y el análisis de datos en contextos académicos.

3.1.1.2 Enfoque Cuantitativo

se aplicaron encuestas estructuradas a estudiantes universitarios de carreras afines a la computación. El propósito es obtener información sobre la frecuencia de uso de herramientas para aprendizaje automático, los problemas que enfrentan en los laboratorios de cómputo y sus opiniones sobre el estado de la infraestructura actual.

Este enfoque mixto permite integrar la comprensión técnica y contextual (cualitativa) con evidencia empírica y medible (cuantitativa), generando una visión completa del impacto de la infraestructura desarrollada (López & Martínez, 2023).

3.1.2. Tipo de Investigación

3.1.2.1 Investigación Aplicada

Busca dar solución a un problema técnico real: la falta de infraestructura adecuada en universidades para el procesamiento eficiente de algoritmos de aprendizaje automático. El objetivo es diseñar e implementar una solución tecnológica que responda a las necesidades prácticas del entorno educativo (Hernández et al., 2023).

3.1.2.2 Investigación Descriptiva

Permite caracterizar el estado actual de los laboratorios de cómputo, las herramientas existentes y la infraestructura disponible en la institución. A través de esta descripción detallada, se identifican fortalezas y debilidades del entorno, sirviendo como base para justificar el diseño de la nueva solución (Ramírez & Ortega, 2023).

3.1.2.3 Investigación Explicativa

Analiza las causas de los problemas técnicos observados (baja eficiencia, tiempos de espera prolongados, recursos no optimizados) y explica cómo una infraestructura distribuida puede corregir estas deficiencias. Este tipo de investigación permite establecer relaciones causa-efecto y sustentar las decisiones de diseño tomadas durante el proyecto (Sampieri et al., 2023).

3.1.2.4 Investigación Documental

Consiste en la revisión de literatura científica, tesis, artículos académicos y manuales técnicos para fundamentar teóricamente la propuesta. Se analiza información actualizada sobre tecnologías de virtualización, orquestación, computación distribuida y frameworks de Machine Learning (González & Pérez, 2024).

3.2. HIPÓTESIS

Hipótesis Principal:

La implementación de un prototipo de infraestructura tecnológica distribuida, optimizada con herramientas especializadas y recursos de alto rendimiento, mejora significativamente el procesamiento y análisis de algoritmos de aprendizaje automático en laboratorios universitarios, al reducir los tiempos de ejecución, incrementar la precisión de los modelos y optimizar el uso de los recursos computacionales.

3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE LAS VARIABLES

3.3.1. Definición de las Variables

Conjunto de herramientas tecnológicas, ya sean hardware o software, que facilitan la ejecución y escalado eficaz de algoritmos de aprendizaje automático. Esto abarca servidores, redes veloces, almacenamiento distribuido, unidades de procesamiento como GPUs y TPUs, y plataformas de nube que facilitan la escalabilidad dinámica de los recursos.

Mejoras en el desempeño informática de los algoritmos de aprendizaje automático, a través de la correcta distribución de recursos tecnológicos. Esto implica disminuir los tiempos de procesamiento, incrementar la exactitud de los modelos y potenciar la habilidad para gestionar grandes cantidades de datos sin perjudicar el desempeño del sistema.

3.3.2. Operacionalización de las variables

Tabla 3. Especificación de indicadores de variables

Definición Operativa de las Variables						
	Variable	**Definición**	**Dimensión**	**Indicador**	**Técnica**	**Instrumento**
Variable independiente	Implementación de Infraestructura para el Procesamiento de Algoritmos de Machine Learning	Conjunto de pasos, recursos y tecnologías utilizados para construir un entorno computacional eficiente que permita el desarrollo y ejecución de algoritmos de Machine Learning.	Recursos computacionales	– Capacidad de CPU, GPU, y RAM instaladas Espacio de almacenamiento disponible	Observación y análisis de sistemas	Especificaciones de hardware y software
Variable dependiente	Optimización del procesamiento de algoritmos	Incremento en la eficiencia y rendimiento de los algoritmos en un entorno de Machine Learning .	Eficiencia operativa	– - Tiempos de ejecución - Capacidad de respuesta	- análisis de rendimiento	informes de rendimiento
			Precisión	- Reducción de errores – Resultados esperados	- Pruebas de validación	- Registros de pruebas.
			reducción de costos	- ahorro en recursos computacionales –	análisis económico	- informes financieros.
			Escalabilidad	capacidad para manejar cargas mayores sin errores	- pruebas de carga	Informes de pruebas y simulaciones

3.4. MÉTODOS UTILIZADOS

El método aplicado en este proyecto es de tipo experimental y tecnológico-aplicado, orientado al diseño y validación de un prototipo funcional de infraestructura distribuida para el procesamiento de algoritmos de Machine Learning en los contextos universitarios. Para lo cual se utilizaron métodos mixtos, para poder combinar los enfoques cualitativos y cuantitativos

3.4.1 Método Inductivo

Se aplicó para analizar los datos cualitativos obtenidos en entrevistas a docentes e investigadores, para poder identificando necesidades comunes, patrones de uso de recursos y las barreras en el acceso a tecnologías en ciencia de datos. Este análisis sirvió como base para definir las características funcionales y técnicas de la infraestructura.

3.4.2 Método Descriptivo

Este siguiente método permitió caracterizar el estado que se encuentran actualmente los laboratorios de cómputo, y así poder detallar sus capacidades, limitaciones y herramientas utilizadas en la enseñanza e investigación con algoritmos. A partir de esta caracterización, se estructuraron los requerimientos técnicos necesarios para la propuesta.

3.4.3 Desarrollo del Prototipo Experimental

El sistema fue desarrollado en fases secuenciales, iniciando con el análisis de requerimientos institucionales, seguido del diseño conceptual de la arquitectura tecnológica. Posteriormente, se implementó un entorno funcional sobre un servidor prestado, utilizando herramientas de código abierto y priorizando el uso de Hadoop como plataforma principal para procesamiento distribuido.

Durante el proceso se configuraron recursos computacionales, se integraron frameworks de Machine Learning y se realizaron pruebas controladas con datos reales y artificiales. Las pruebas permitieron evaluar aspectos tales como el rendimiento, escalabilidad, facilidad de uso y utilidad académica.

3.4.4 Validación y Ajustes

El prototipo fue evaluado por estudiantes mediante pruebas prácticas, recolección de métricas del sistema y encuestas. Con base en la retroalimentación recibida, se

realizaron ajustes técnicos, se documentaron las configuraciones y se elaboraron manuales de uso. Finalmente, se valoró el impacto de la infraestructura en términos de reducción de tiempos de procesamiento, aprovechamiento de recursos y replicabilidad para otros laboratorios.

IV. RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

4.1.1. Análisis de encuesta

Pregunta 1

¿En su universidad existe una infraestructura básica para el procesamiento y análisis de algoritmos dirigido para Machine Learning?

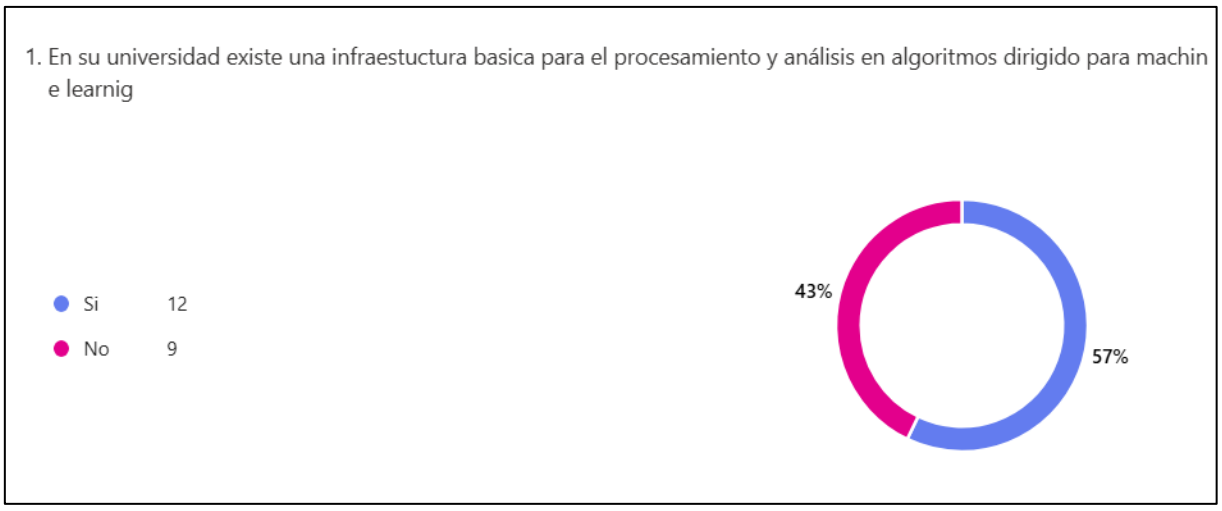


Figura 1. Pregunta 1

El resultado final de la encuesta demuestra que el 57% de los encuestadores saben que su universidad cuenta con una infraestructura básica para el procesamiento y análisis en algoritmos, pero se ve la división grande que es del 43% dicen que no cuenta con una infraestructura básica, esto refleja que si bien su universidad cuenta con unos recursos tecnológicos mínimos aún existe una porción mínima que no cuenta con una infraestructura en entornos universitarios

Pregunta 2

Los recursos tecnológicos actuales que tienen le permiten aprender machine learning de forma efectiva

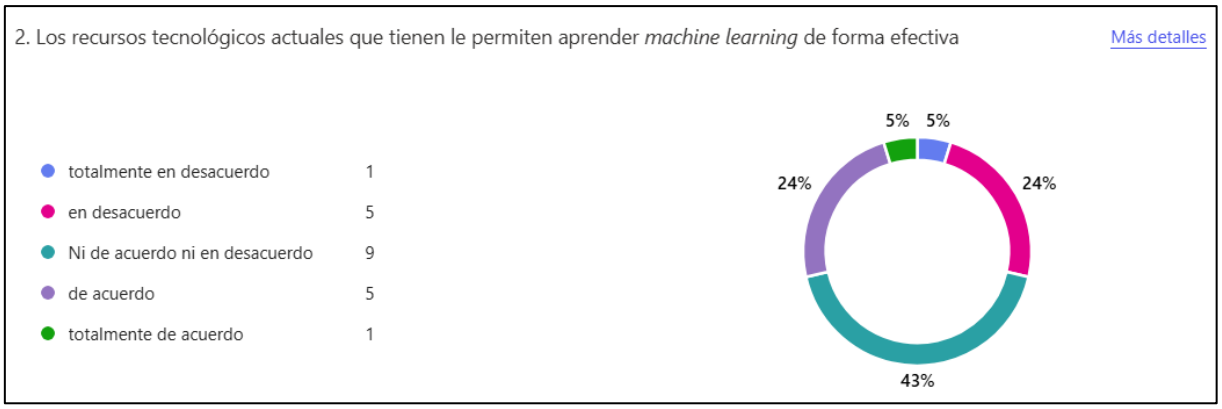


Figura 2. Pregunta 2

En el siguiente resultado los estudiantes dieron su punto de vista de que si creen que las tecnologías actuales permiten aprender machine learning estos resultados que el 43% se mantiene de forma neutral mientras que un 24% supo manifestar que están de acuerdo que si existe tecnología suficiente para aprender mientras que los que están totalmente en de acuerdo y en desacuerdo se mantiene en un 5% cada uno esto puede indicar que podemos encontrar recursos disponibles de estas herramientas

Pregunta 3

¿Es importante tener acceso a herramientas específicas para practicar y desarrollar proyectos de machine learning?

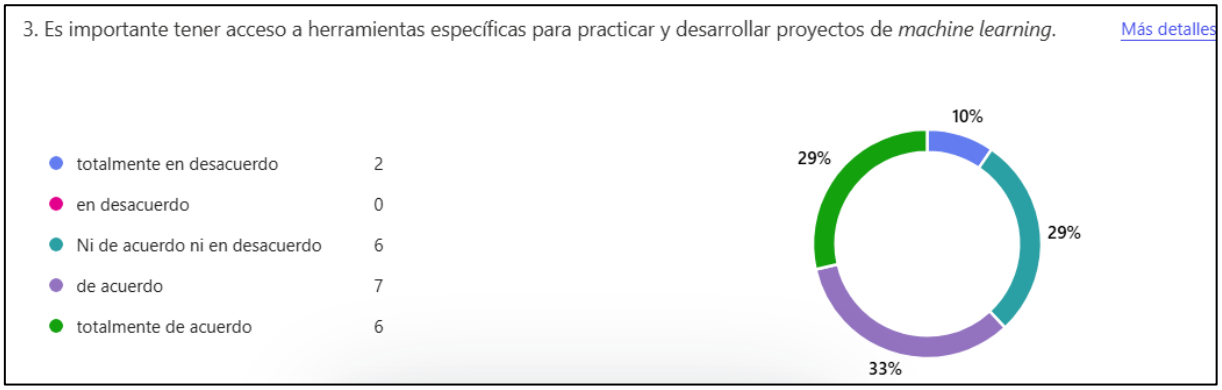


Figura 3. Pregunta 3

En la siguiente pregunta los resultados indican una inclinación favorable en estar de acuerdo con un 33% y un 29% totalmente de acuerdo de que se debe tener acceso a herramientas específicas para la práctica y el desarrollo mientras que un 29% se mantiene de forma neutral y por ultimo tenemos que un 10% está totalmente en desacuerdo, esto indica que cada uno de los estudiantes tienen diferente

percepción acerca la relevancia que tienen a las herramientas especializadas para el desarrollo practico en el campo de machine learning

Pregunta 4

¿El aprendizaje de machine learning están limitado por la falta de infraestructura adecuada como computadoras potentes o programas especializados?

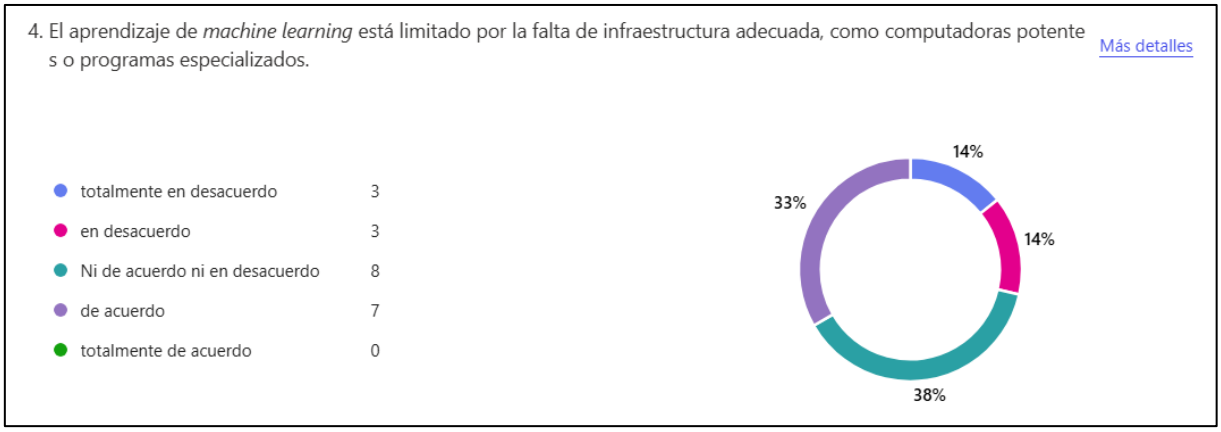


Figura 4. Pregunta 4

En esta pregunta los resultados nos muestran que un 38% de los encuestados se mantienen en forma neutral que un 33% está de acuerdo y que un 14% de los encuestados se mantienen en desacuerdo y totalmente en desacuerdo cada uno esto nos refleja que si es necesario contar con una infraestructura para obtener un aprendizaje efectivo en el campo de machine learning

Pregunta 5

¿Los recursos gratuitos disponibles para aprender machine learning son suficientes para empezar a practicar?

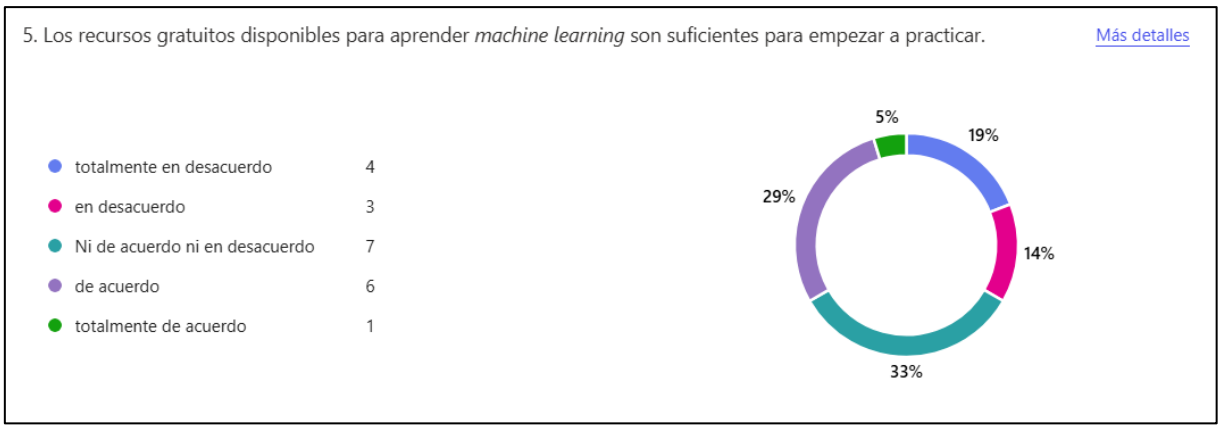


Figura 5. Pregunta 5

En los resultados muestran nuevamente que un 33% se mantiene de forma neutral que un 29% está de acuerdo y que un 5% en totalmente de acuerdo esto indica que los recursos gratuitos para el aprendizaje de machine learning son suficientes para poder empezar a practicar sin embargo un 19% está en totalmente en desacuerdo y un 14% en desacuerdo esto muestra los diferentes tipos de perspectivas que tienen cada uno de los encuestadores

Pregunta 6

¿Considera que las computadoras que utilizas suelen ser rápidas y eficientes para trabajar con algoritmos básicos de machine learning?

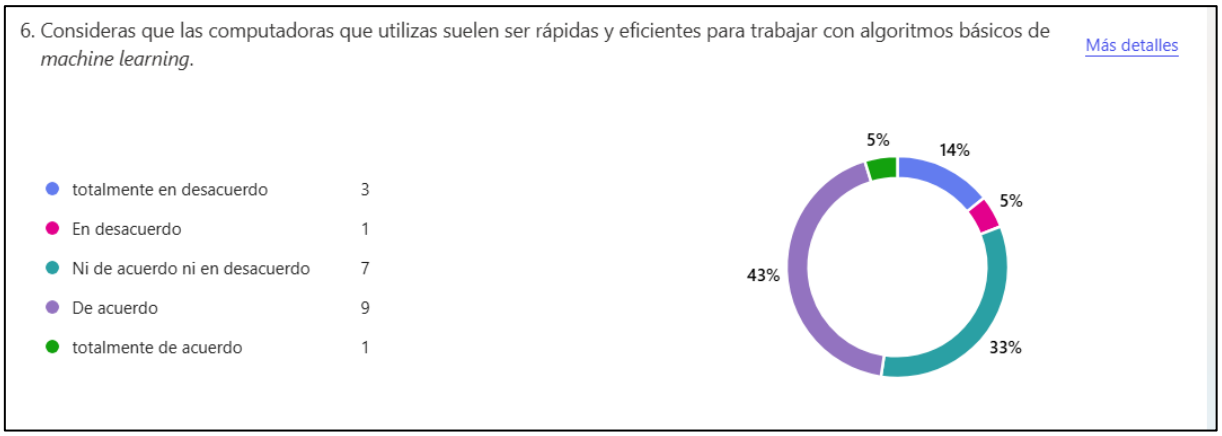


Figura 6. Pregunta 6

En estos resultados indican que el 43% está de acuerdo y un 5% en totalmente de acuerdo esto nos indica que casi el 48% de los encuestadores cuentan con computadores rápidos para el trabajo en el campo de machine learning en cambio un 14% está en totalmente en desacuerdo y un 5% en desacuerdo significan que no cuentan con un computador lo suficientemente rápido para poder realizar trabajos de este campo

Pregunta 7

¿Te gustaría tener acceso a recursos más especializados para realizar proyectos más avanzados de machine learning?

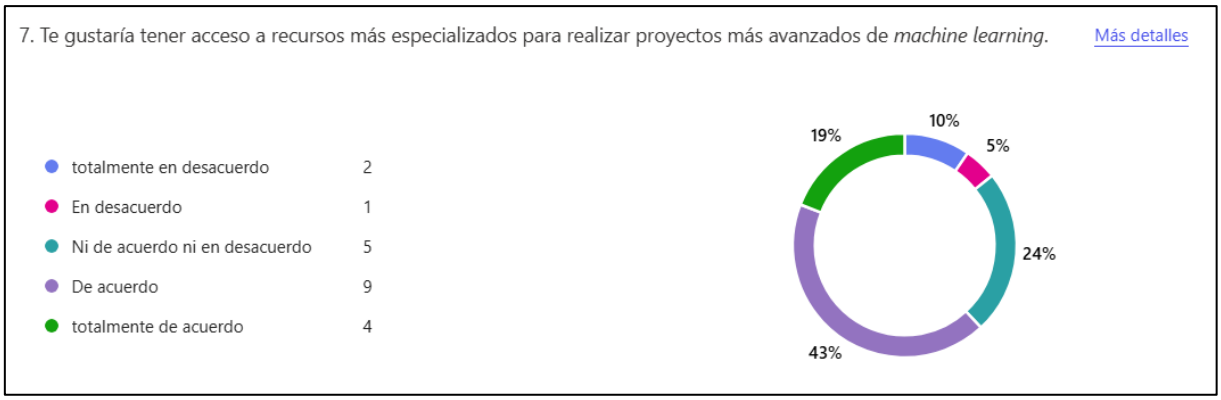


Figura 7. Pregunta 7

Los resultados indican una clara inclinación a la positividad debido a que cuenta con un 43% en de acuerdo y un 19% totalmente de acuerdo lo que suma un 62% más de la mitad de los encuestadores que les gustaría tener recursos especializados para realizar proyectos de machine learning

Pregunta 8

¿Crees que contar con mejores herramientas tecnológicas te ayudaría a entender y aplicar conceptos de machine learning?

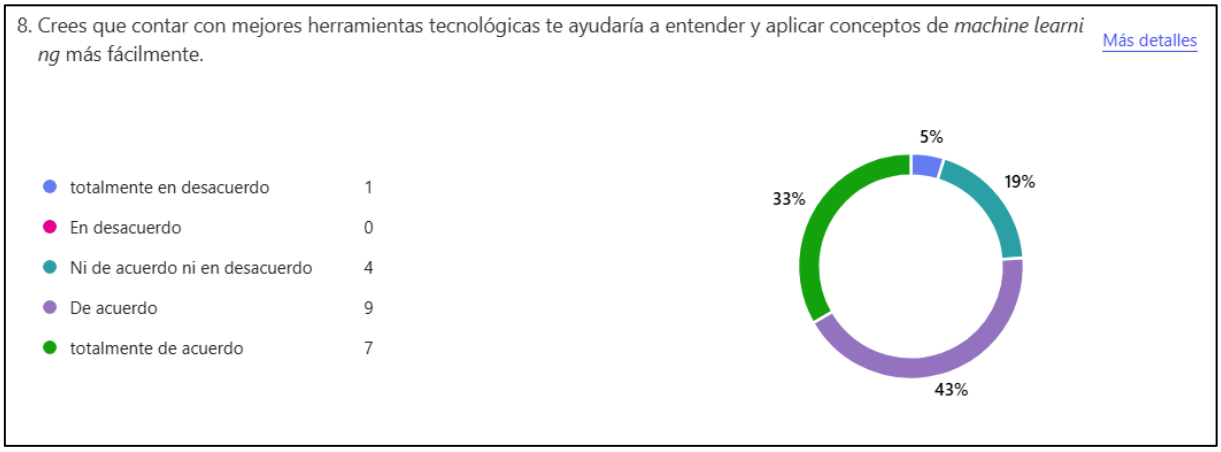


Figura 8. Pregunta 8

Los resultados muestran una tendencia favorable, con un 43% que está de acuerdo y un 19% que está totalmente de acuerdo, sumando así un 62% de respuestas positivas. Esto demuestra que la mayoría de los encuestados tiene el interés de contar con recursos más especializados que les permitan desarrollar proyectos más avanzados en *machine learning*

Pregunta 9

¿Considero que sería útil contar con un espacio dedicado para practicar proyectos de machine learning?

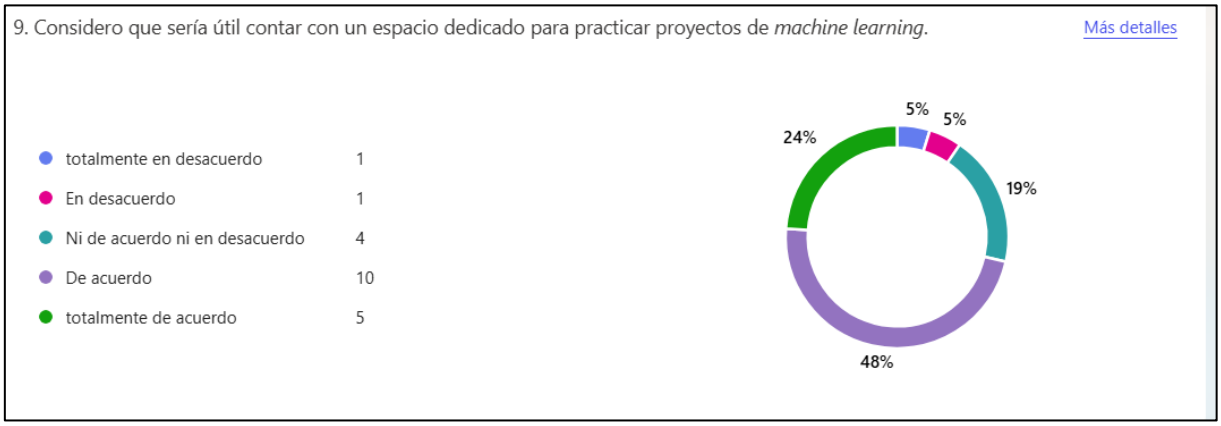


Figura 9. Pregunta 9

Los resultados reflejan una alta aceptación por parte de los encuestados, ya que un 48% está de acuerdo y un 24% totalmente de acuerdo, sumando un 72% de opiniones positivas. Esto indica que una gran mayoría considera que sería útil contar con un espacio dedicado exclusivamente para practicar proyectos de machine learning, lo cual resalta la necesidad de crear ambientes adecuados que fomenten el aprendizaje práctico dentro de la universidad.

Pregunta 10

¿Encontrar herramientas adecuadas para machine learning es un desafío constante en mi proceso de aprendizaje?

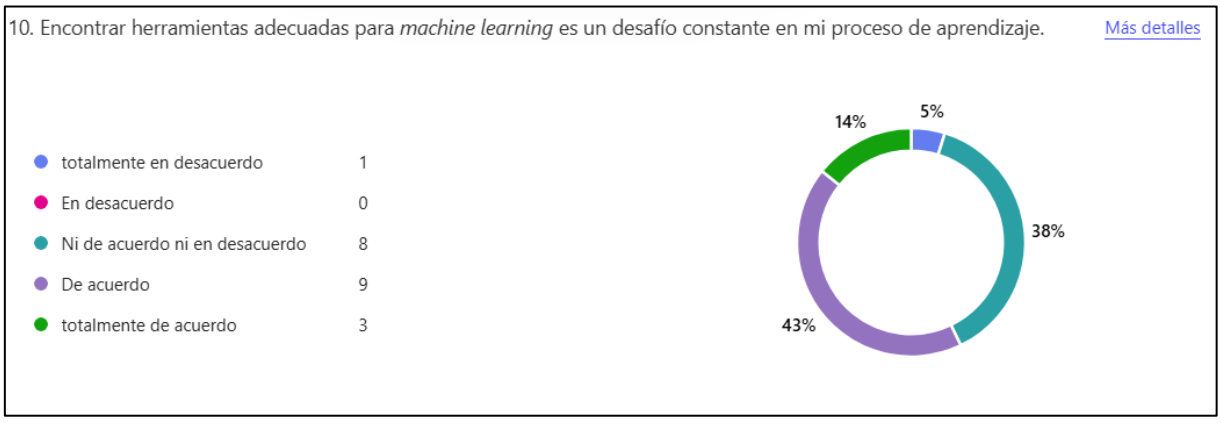


Figura 10. Pregunta 10

Los resultados muestran que un 43% está de acuerdo y un 14% totalmente de acuerdo, lo que suma un 57% que considera que encontrar herramientas adecuadas para *machine learning* es un desafío constante en su proceso de aprendizaje.

Además, un 38% se mantuvo neutral y solo un 5% estuvo totalmente en desacuerdo. Esto indica que más de la mitad de los encuestados enfrenta dificultades al buscar recursos apropiados, lo que evidencia una problemática real que afecta el avance y la continuidad en el aprendizaje de esta área tecnológica.

4.2. PROPUESTA

Esta propuesta nace a partir de observar las limitaciones que enfrentan hoy en día muchos laboratorios de cómputo en las universidades. Durante el desarrollo del trabajo se identificó que tareas como el procesamiento de datos en grandes cantidades y la ejecución de algoritmos de machine learning necesitan una infraestructura que sea más flexible, escalable y que ofrezca mejores tiempos de respuesta.

Ante esta situación, se propuso diseñar e implementar un sistema distribuido enfocado en facilitar el análisis de modelos de aprendizaje automático. La idea principal fue aprovechar mejor los recursos tecnológicos disponibles, optimizar el tiempo que toma entrenar los modelos y crear un entorno de trabajo más eficiente y funcional para estudiantes e investigadores que trabajan con ciencia de datos.

Para probar que esta solución puede funcionar, se utilizó una laptop personal y se configuraron dos máquinas virtuales que actuaron como nodo maestro y nodo esclavo. Con esta configuración sencilla y económica, fue posible demostrar que el sistema funciona en condiciones reales y con pocos recursos. El desarrollo se organizó por etapas, incluyendo la instalación del software base, la configuración de herramientas de análisis distribuidas y las pruebas para comprobar su funcionamiento.

El sistema usa tecnologías que permiten el procesamiento en paralelo, almacenamiento distribuido y también herramientas que ayudan a hacer seguimiento de los experimentos realizados. Esto hace posible que los modelos se ejecuten de manera más eficiente y que diferentes usuarios puedan colaborar en los proyectos. Como resultado, se logró crear una solución tecnológica que es estable, fácil de replicar y adecuada para lo que hoy demandan los espacios académicos en temas de aprendizaje automático.

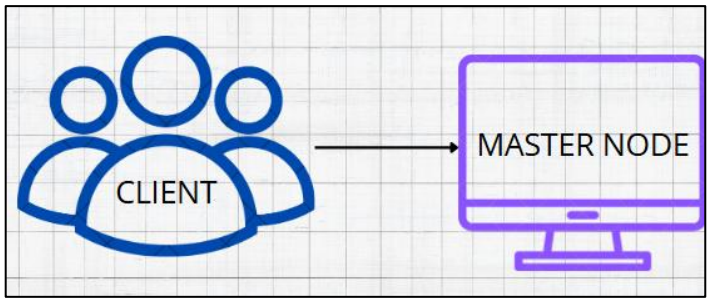


Figura 11. Modelo de conexión inicial en un sistema distribuido

4.2.1 Estudio de Factibilidad

Este estudio de factibilidad evalúa la viabilidad técnica, económica, organizacional y operativa de una infraestructura tecnológica propuesta como modelo replicable, diseñada para ser implementada por instituciones académicas interesadas en optimizar el procesamiento de datos en entornos de aprendizaje automático. No se desarrolla dentro de una institución específica, sino que plantea una solución adaptable a diferentes universidades o centros de investigación.

Estudio de Factibilidad

Tema: Infraestructura para el procesamiento de análisis en algoritmos dirigido para Machine Learning

4.2.1.1 Factibilidad Organizacional

Esta propuesta está orientada a entornos académicos como universidades o centros de investigación que deseen implementar por cuenta propia una infraestructura distribuida para Machine Learning. El diseño del proyecto responde a necesidades comunes en instituciones educativas: autonomía investigativa, acceso a tecnologías emergentes y demanda por recursos computacionales avanzados.

Misión: Proveer una solución tecnológica escalable, segura y accesible para facilitar el entrenamiento y análisis de algoritmos de aprendizaje automático en instituciones educativas.

Visión: Ser un modelo replicable de infraestructura académica distribuida, adaptable a distintos contextos universitarios, que fomente la investigación y formación en inteligencia artificial.

Organización del Proyecto:

El desarrollo de la infraestructura seguirá un modelo basado en las siguientes fases:

- Diagnóstico de necesidades: Identificación de requerimientos técnicos, tecnológicos y académicos necesarios para la implementación de la infraestructura.
- Diseño de la arquitectura tecnológica: Selección de hardware, software y herramientas, considerando escalabilidad, compatibilidad y rendimiento.
- Implementación de la infraestructura base: Configuración de servidores, instalación de sistemas operativos, redes y plataformas de almacenamiento.
- Integración de herramientas de Machine Learning: Incorporación de frameworks, entornos interactivos y plataformas distribuidas para el procesamiento de datos.
- Pruebas, documentación y capacitación: Evaluación del rendimiento, generación de manuales técnicos, y formación a los usuarios académicos.

4.2.1.2 Factibilidad Técnica

La factibilidad técnica considera los recursos mínimos que una institución educativa podría requerir para implementar el modelo propuesto. No implica que estos recursos estén actualmente disponibles, sino que se describen como referencia para quienes deseen adoptarlo.

Recursos Disponibles:

- Uso de dos máquinas virtuales configuradas con CentOS 9 para simular un entorno distribuido.
- Instalación y configuración de Apache Hadoop y Apache Spark.
- Utilización de PySpark como herramienta principal para pruebas de procesamiento distribuido.
- Recursos computacionales limitados, pero suficientes para pruebas de concepto y validación funcional a pequeña escala.

Requerimientos Técnicos:

- Instalación del sistema operativo CentOS 9 en ambas máquinas virtuales.
- Configuración manual de la red, roles maestro-esclavo, y variables de entorno para Hadoop y Spark.
- Ejecución de tareas básicas de procesamiento distribuido utilizando scripts en PySpark.

La evaluación técnica concluye que, aunque se trata de un entorno limitado y simulado, el modelo es funcional para fines educativos y puede escalarse en caso de implementación real por parte de instituciones interesadas.

4.2.1.2.1 Cuadro comparativo Técnico-Económico de arquitecturas distribuidas.

Los costos presentados a continuación son estimaciones las cuales son basadas en licencias libres o académicas orientadas a que las instituciones interesadas podrían evaluarlo en su implementación según sus propios presupuestos.

Tabla 4. Comparación de aspectos técnicos y económicos de arquitecturas distribuidas

Análisis de Costos y Desempeño en Arquitecturas Distribuidas

Arquitectura/ Herramienta	Características Técnicas	Ventajas	Desventajas	Aplicación Académica	Precio Estimado (USD)
Apache Hadoop	HDFS + MapReduce para procesamiento batch	Escalable, confiable	Requiere soporte técnico	Procesamiento de datos históricos en laboratorios	$1.000 – $2.500
Apache Spark	Procesamiento en memoria, incluye MLlib	Rápido, ideal para grandes volúmenes de datos	Mayor requerimiento de memoria	Procesamiento distribuido, clases de Big Data	$1.500 – $3.000
Docker + Kubernetes	Contenedores + gestión escalable de servicios	Versatilidad, modularidad	Requiere configuración experta	Infraestructura reproducible en investigación	$1.000 – $2.000
Ray	Biblioteca Python para tareas paralelas	Ideal para ML, rápida implementación	Comunidad y documentación más limitada	Experimentación ligera con modelos ML	$500 – $1.500
Dask	Escala operaciones tipo Pandas en paralelo	Muy ligero y didáctico	No recomendado para proyectos grandes	Clases prácticas y prototipos simples	$300 – $1.000
TensorFlow Distributed	Entrenamiento distribuido sobre GPU/TPU	Gran rendimiento con Deep Learning	Requiere hardware potente	Laboratorios de IA y redes neuronales	$2.500 – $5.000
OpenStack (IaaS)	Plataforma de nube privada escalable	Virtualización completa de recursos	Infraestructura costosa y compleja	Centros de datos universitarios avanzados	$5.000 – $15.000

4.2.1.3 Factibilidad Económica

El proyecto fue desarrollado de forma individual, sin apoyo institucional, utilizando una computadora personal Asus TUF Gaming con procesador Intel Core i5 de 10ª generación, tarjeta gráfica GeForce GTX 1650Ti y 16 GB de RAM. La implementación se realizó mediante dos máquinas virtuales con CentOS 9, utilizando software libre como Hadoop, Spark y PySpark. No se incurrió en gastos adicionales.

Tabla 5. Evaluación económica

Análisis de Costos y Beneficios			
Descripción	**Cantidad**	**Costo Unitario**	**Costo Total**
Computadora Asus TUF Gaming (i5 10ª Gen, GTX 1650Ti, 16GB RAM	1	$1,500	$1,500
Almacenamiento externo (de alto rendimiento 1TB)	1	$100	$100
Instalación de software y configuración de VMs	1	$0	$0
Sistema operativo(libre)	1 VMs	$0	$0
Hadoop, Spark y PySpark (software libre)	1	$0	$0
Total, Estimado			$1,600

4.2.1.4 Factibilidad Operativa

Se considera la factibilidad operativa desde el punto de vista de las instituciones educativas que deseen adoptar el modelo propuesto. Actualmente, muchos laboratorios universitarios enfrentan limitaciones técnicas que esta infraestructura ayudaría a solventar. La propuesta ofrece un entorno ideal de referencia, adaptable a diferentes condiciones técnicas.

4.2.2 Propuesta de implementación

Para facilitar la adopción del modelo propuesto en instituciones académicas interesadas, se sugiere una implementación en cinco fases consecutivas. Esta estructura no se basa en metodologías ágiles, sino en una planificación tradicional adecuada para entornos educativos y administrativos.

- **Fase 1: Diagnóstico Inicial:** Revisión de las necesidades académicas, disponibilidad de infraestructura existente, y requerimientos específicos de procesamiento para Machine Learning.
- **Fase 2: Preparación del Entorno Tecnológico:** Selección e instalación del sistema operativo, configuración de red segura, implementación de plataformas de virtualización y contenedores (como Docker).
- **Fase 3: Integración de Herramientas:** Instalación de entornos interactivos (Jupyter), herramientas de procesamiento distribuido (Apache Spark, PySpark) y librerías de aprendizaje automático (Scikit-learn).
- **Fase 4: Validación Técnica:** Ejecución de pruebas funcionales y de rendimiento con datasets de prueba. Evaluación del entorno por parte de usuarios académicos.

- **Fase 5: Documentación y Capacitación:** Elaboración de guías de uso, capacitación básica a docentes y estudiantes, y recomendaciones para futuras réplicas en otros laboratorios.

4.2.3 Recursos a Utilizar

Los recursos listados no representan activos actuales del autor del proyecto, sino una sugerencia de tecnologías y talentos requeridos para que cualquier institución pueda implementar esta propuesta.

- **Tecnológicos:** Hadoop, Apache Spark, PySpark, máquina virtual con sistema operativo Linux (CentOS o Ubuntu).
- **Humanos:** Trabajo individual, con apoyo teórico de fuentes académicas y guías técnicas disponibles en línea.
- **Logísticos:** Máquina virtual local configurada para pruebas básicas; no se utilizó infraestructura física institucional ni servicios en la nube.

4.2.4 Beneficios Esperados

Los beneficios esperados están proyectados para aquellas universidades o centros de investigación que adopten esta infraestructura. La intención es facilitar la transformación tecnológica en el ámbito educativo.

- Mejora en el rendimiento del entrenamiento de modelos.
- Reducción de tiempos de procesamiento.
- Entorno académico innovador y colaborativo.
- Promoción de proyectos interdisciplinarios.
- Infraestructura replicable para otras instituciones.
- Fortalecimiento de la formación en inteligencia artificial.

4.3 DISCUSIÓN

La implementación de una infraestructura distribuida basada en tecnologías de código abierto, como Hadoop y Apache Spark, ha demostrado ser una alternativa efectiva frente a las limitaciones técnicas presentes en muchos laboratorios universitarios. A través del entorno desarrollado en máquinas virtuales, fue posible simular un clúster funcional que permitió validar los beneficios de este tipo de arquitectura en condiciones reales de uso académico.

Durante las pruebas realizadas, se observó que Apache Spark, al ejecutar procesos en memoria, ofrece tiempos de respuesta significativamente mejores en comparación con tecnologías tradicionales, siempre que la configuración del sistema, la asignación de recursos y la estructura de los datos sean adecuadas. Etapas como la limpieza de datos, el análisis exploratorio y el entrenamiento de modelos fueron notoriamente más rápidas cuando se optimizó correctamente el entorno, en especial en lo referente al uso de memoria, número de núcleos virtuales y tamaño de particiones en HDFS.

El uso de HDFS como sistema de almacenamiento distribuido también resultó beneficioso, ya que permitió organizar datasets grandes y garantizar su disponibilidad a través de los nodos del clúster. La interoperabilidad entre Spark y Hadoop reforzó la eficiencia del sistema, logrando una ejecución fluida en pruebas con volúmenes moderados de datos académicos.

Una de las principales fortalezas de esta propuesta es su capacidad de escalar horizontalmente. A pesar de haber sido validada en una laptop con dos máquinas virtuales, el diseño permite añadir nuevos nodos fácilmente, lo cual representa una solución accesible para universidades que desean modernizar sus laboratorios sin inversiones excesivas. Esto aprovecha los recursos existentes mediante software libre y tecnologías probadas en la industria.

Desde una perspectiva educativa, la infraestructura propuesta proporciona un entorno realista para que estudiantes y docentes trabajen con datos auténticos y experimenten con algoritmos de Machine Learning en un entorno distribuido. Esta experiencia fortalece competencias clave como la ejecución paralela, la trazabilidad de experimentos y la optimización de modelos en contextos reales.

El prototipo, además, es replicable y adaptable para otras instituciones educativas. Gracias a su estructura modular y al uso de tecnologías estandarizadas, puede ser reproducido con facilidad en diferentes escenarios académicos. Esto abre la posibilidad a colaboraciones interdisciplinarias e impulsa la relación entre la investigación universitaria y el desarrollo tecnológico.

En conjunto, los resultados obtenidos validan la propuesta tanto a nivel técnico como formativo. La evidencia práctica demuestra que, con una correcta configuración, el uso de tecnologías distribuidas como Hadoop y Spark no solo mejora el rendimiento

en tareas clave del Machine Learning, sino que también representa un paso concreto hacia la transformación digital de los espacios educativos.

4.4 DISEÑO

La infraestructura distribuida propuesta se construye utilizando Hadoop y Apache Spark para procesar grandes volúmenes de datos de manera eficiente, flexible y escalable, en un entorno adecuado para el aprendizaje y la investigación académica.

1. **Hadoop como plataforma de almacenamiento distribuido (HDFS)**
    - **HDFS** es el sistema de almacenamiento clave utilizado en la infraestructura, diseñado para distribuir los datos entre varios nodos. Esto permite que grandes datasets sean gestionados de forma eficiente, mejorando la accesibilidad y la disponibilidad.
    - **Ventajas**: HDFS distribuye los datos de manera que no se depende de un solo servidor físico, mejorando la tolerancia a fallos y asegurando la escalabilidad conforme se aumentan los datos.
2. **Apache Spark como motor de procesamiento en memoria**
    - Apache Spark actúa como motor de procesamiento en memoria, lo que optimiza la ejecución de tareas complejas como el preprocesamiento de datos y el entrenamiento de modelos de *machine learning*.
    - Ventajas: Gracias a su capacidad de procesar datos en memoria y su naturaleza distribuida, Spark mejora considerablemente el rendimiento, especialmente para tareas como análisis exploratorio y entrenamiento de modelos utilizando MLlib, su librería de *machine learning*.
3. **Escalabilidad horizontal**
    - Esta infraestructura permite el escalado horizontal de manera sencilla: si las necesidades de procesamiento aumentan, es posible agregar más nodos al clúster sin grandes complicaciones, lo que otorga flexibilidad y adaptabilidad a las universidades que necesiten ampliar su infraestructura conforme crecen sus proyectos.
4. **Acceso a Jupyter Notebooks con acceso limitado**

- El sistema utiliza Jupyter Notebooks como plataforma interactiva para la ejecución de código y análisis de datos. A través de una contraseña simple (hadoop), se permite el acceso a los notebooks, lo que facilita la interacción con los datos y el desarrollo de proyectos de *machine learning*.
- Seguridad: Aunque la contraseña predeterminada no es la más segura, se puede mejorar mediante configuraciones adicionales de seguridad en el servidor, como cambio de contraseñas, uso de HTTPS o la implementación de un sistema de autenticación multifactorial.

5. **Tecnología de clústeres y gestión de recursos**
   - La infraestructura está diseñada para aprovechar las capacidades de clústeres distribuidos, permitiendo que el procesamiento de datos y la ejecución de modelos de *machine learning* se distribuyan entre varios nodos de cómputo.
   - **Ventaja**: Esto facilita la ejecución de análisis de datos a gran escala, optimizando el uso de recursos sin necesidad de contar con servidores dedicados de alto rendimiento.
6. **Uso de tecnologías de código abierto**
   - Hadoop y Apache Spark son herramientas de código abierto, lo que no solo reduce los costos de implementación, sino que también permite que los estudiantes y docentes tengan acceso a software sin licencias costosas.
   - Ventaja: Al ser tecnologías libres, cualquier institución puede adaptarlas a sus necesidades sin preocuparse por restricciones comerciales, fomentando una educación inclusiva y accesible.

4.5 DESARROLLO

La infraestructura fue implementada utilizando Hadoop para el almacenamiento distribuido mediante HDFS, lo que facilita la gestión eficiente de grandes volúmenes de datos. Para el procesamiento, se empleó Apache Spark, aprovechando su capacidad de procesamiento en memoria y su biblioteca MLlib para realizar tareas de preprocesamiento y entrenamiento de modelos de *machine learning*.

Instalación de hadoop

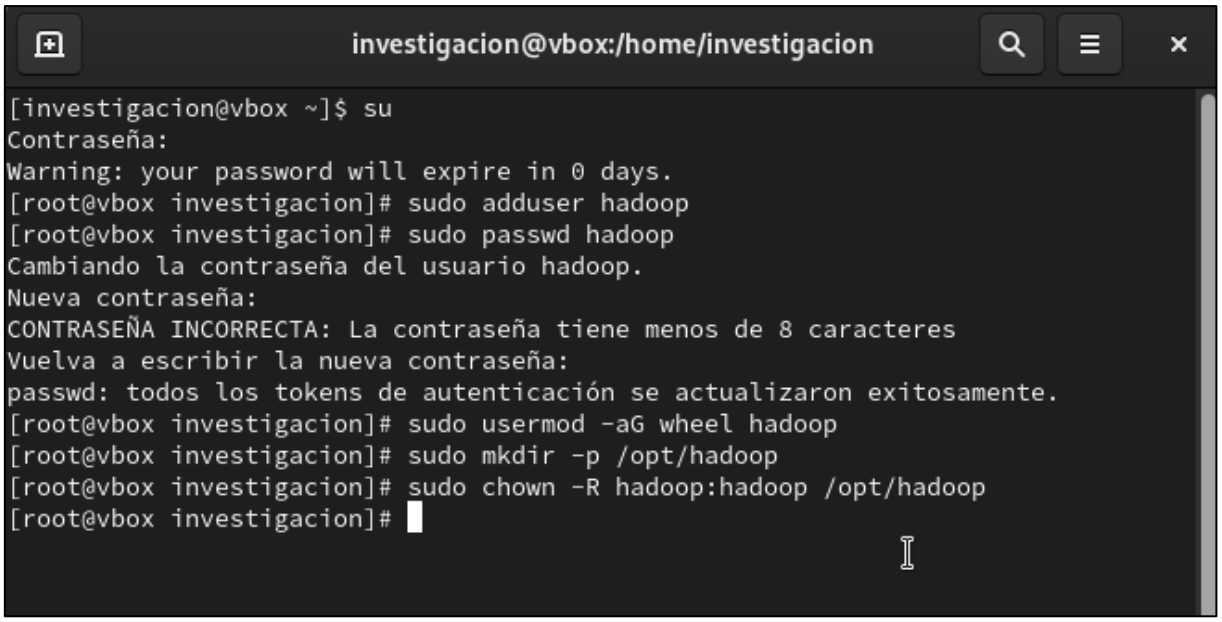


Figura 12. Configuración de un usuario "hadoop"

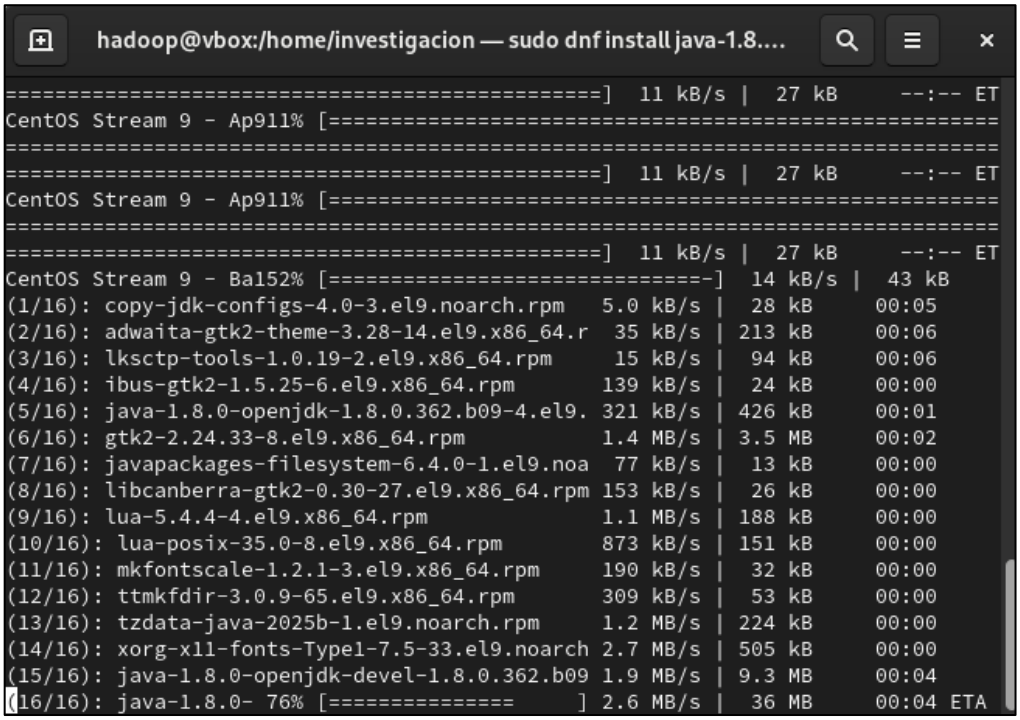


Figura 13. Instalación del JDK 1.8.0

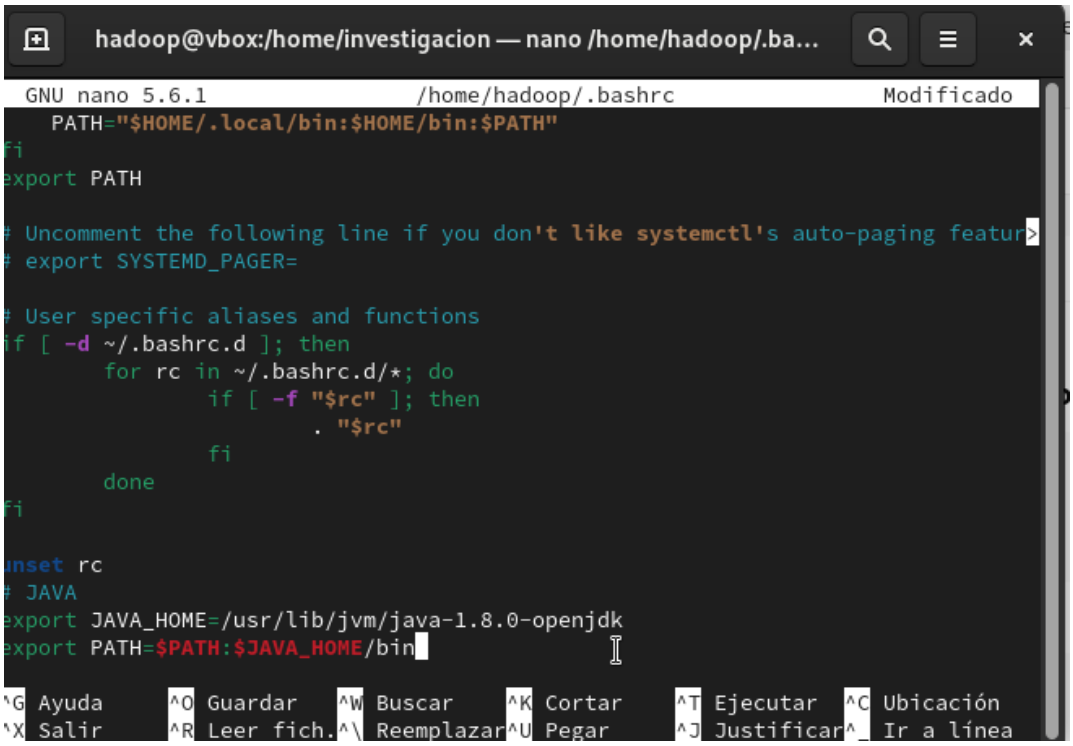


Figura 14. Configuración del bash del JDK

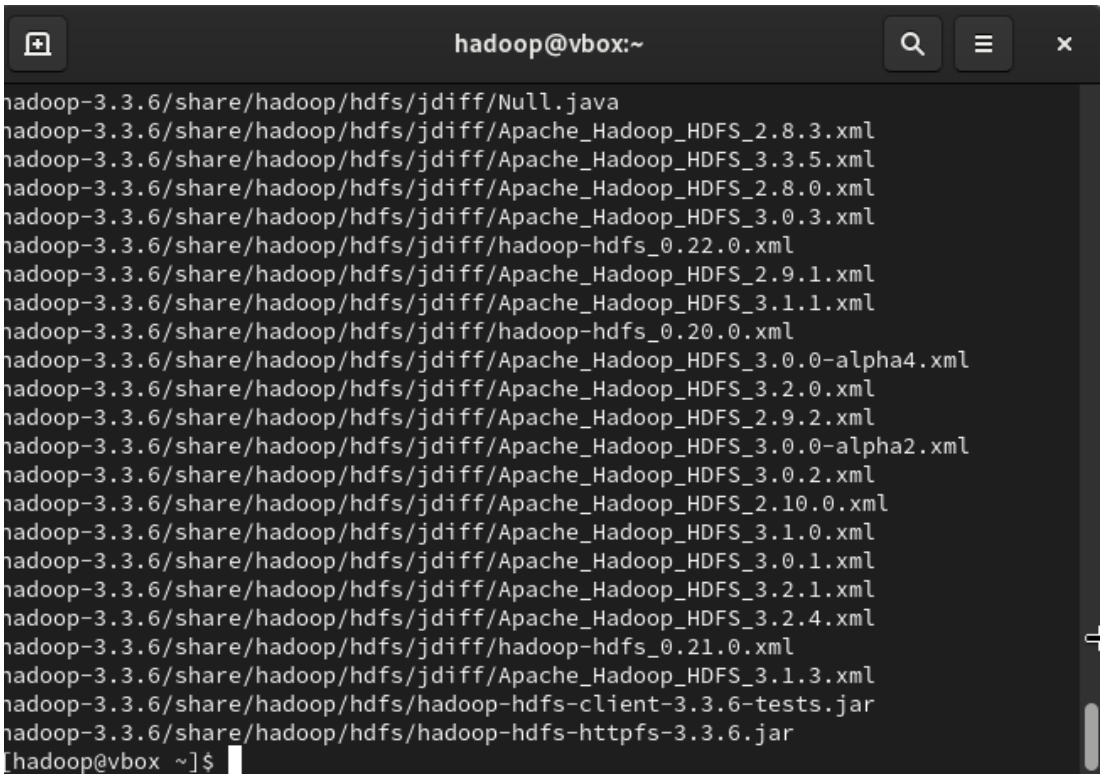


Figura 15. Descarga de hadoop

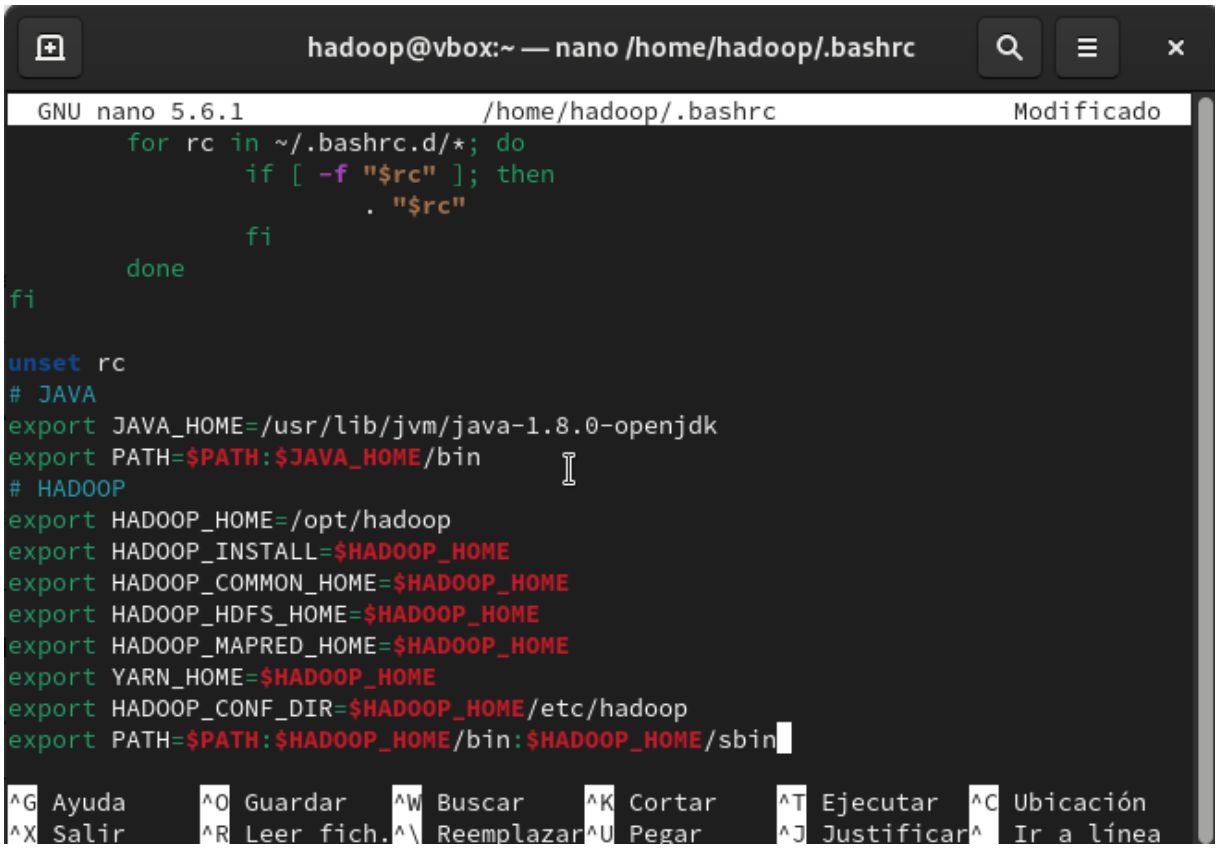


Figura 16. Configuración del bash de hadoop

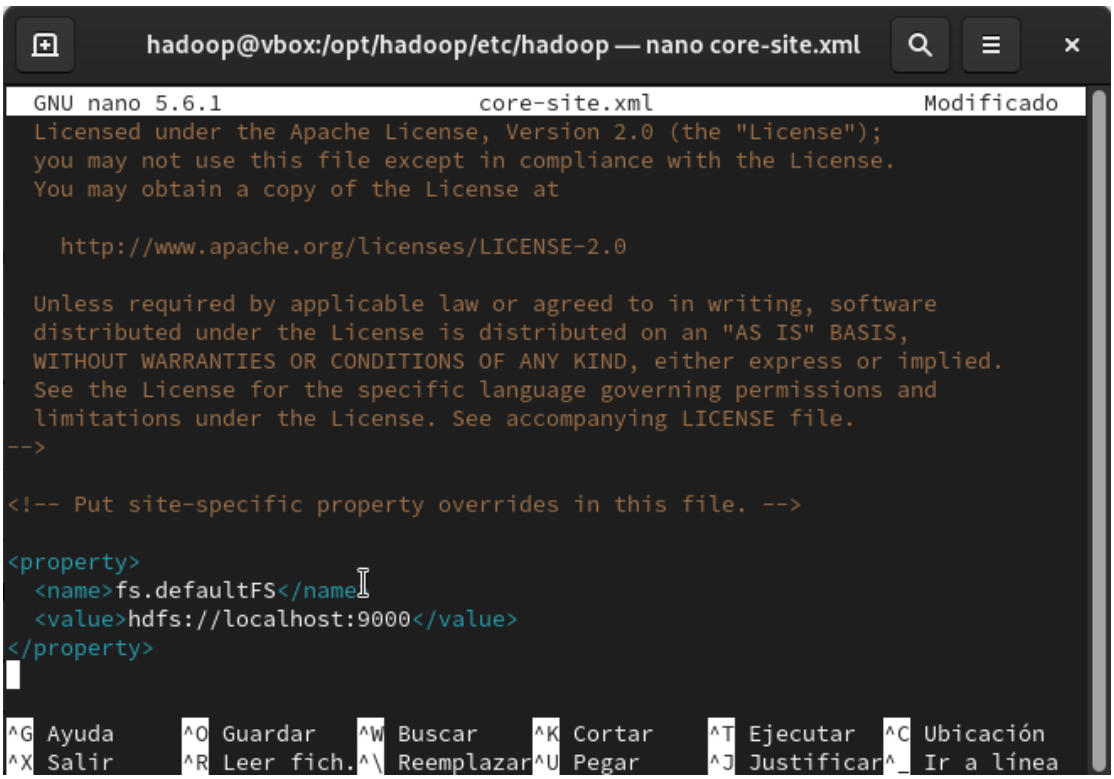


Figura 17. Configuración del nano core-site.xml

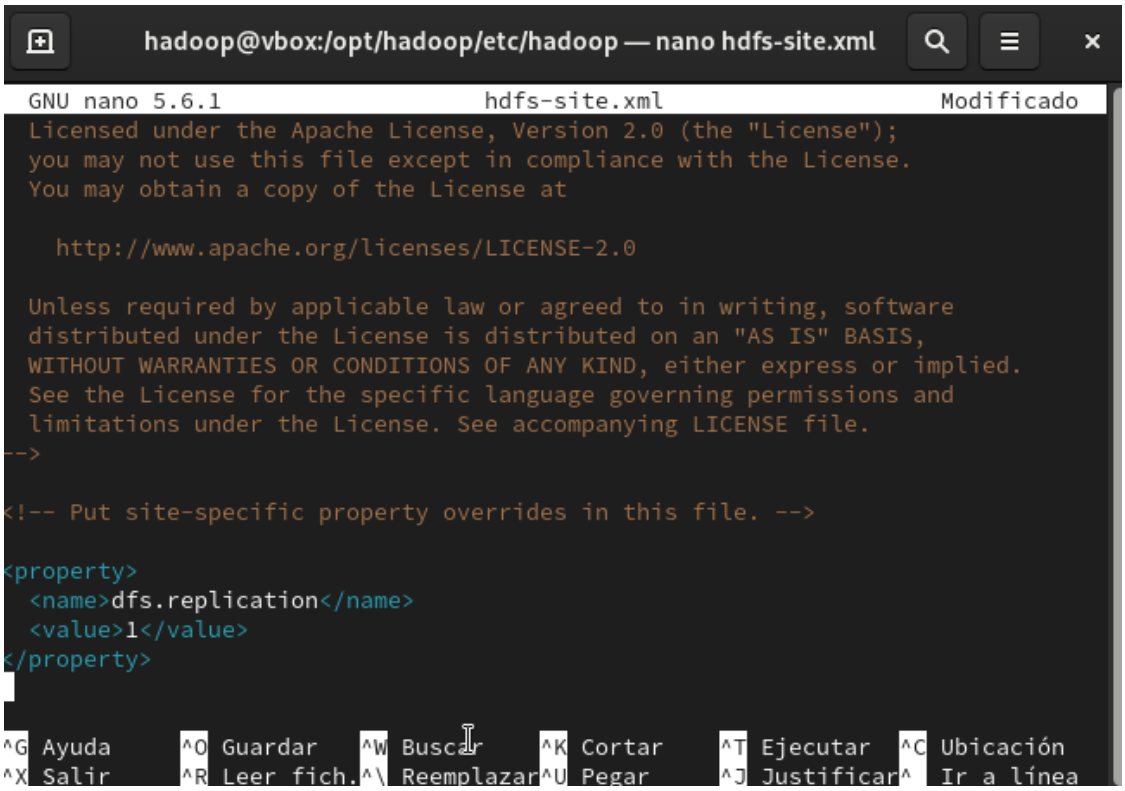


Figura 18. Configuración del nano hdfs-site.xml

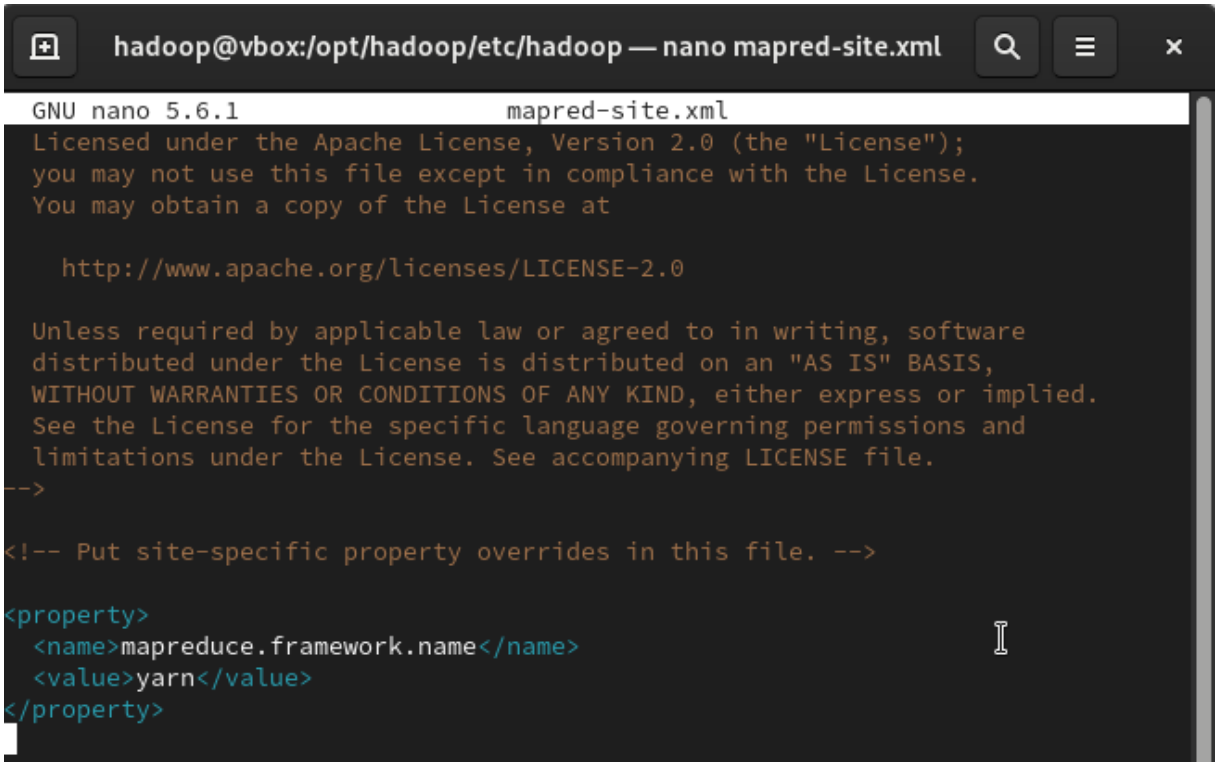


Figura 19. Configuración del nano mapred-site.xml

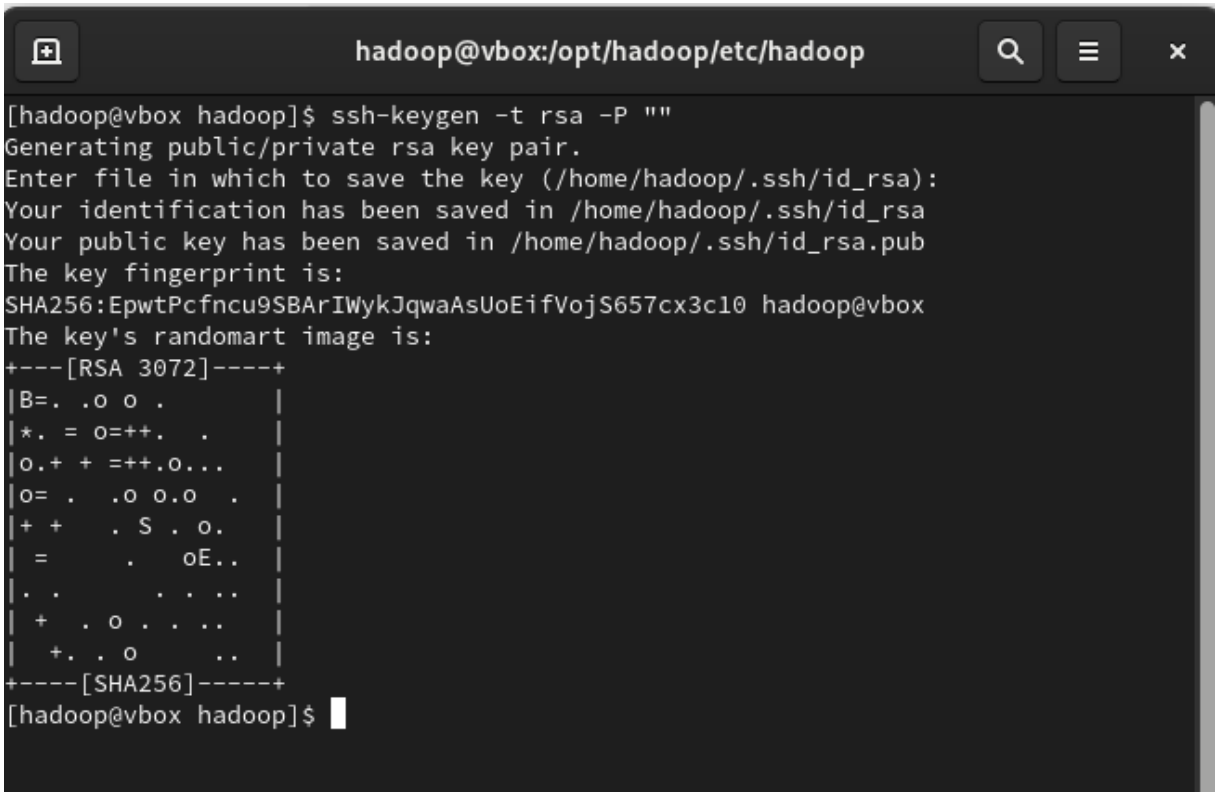


Figura 20. Creación de una clave SSH

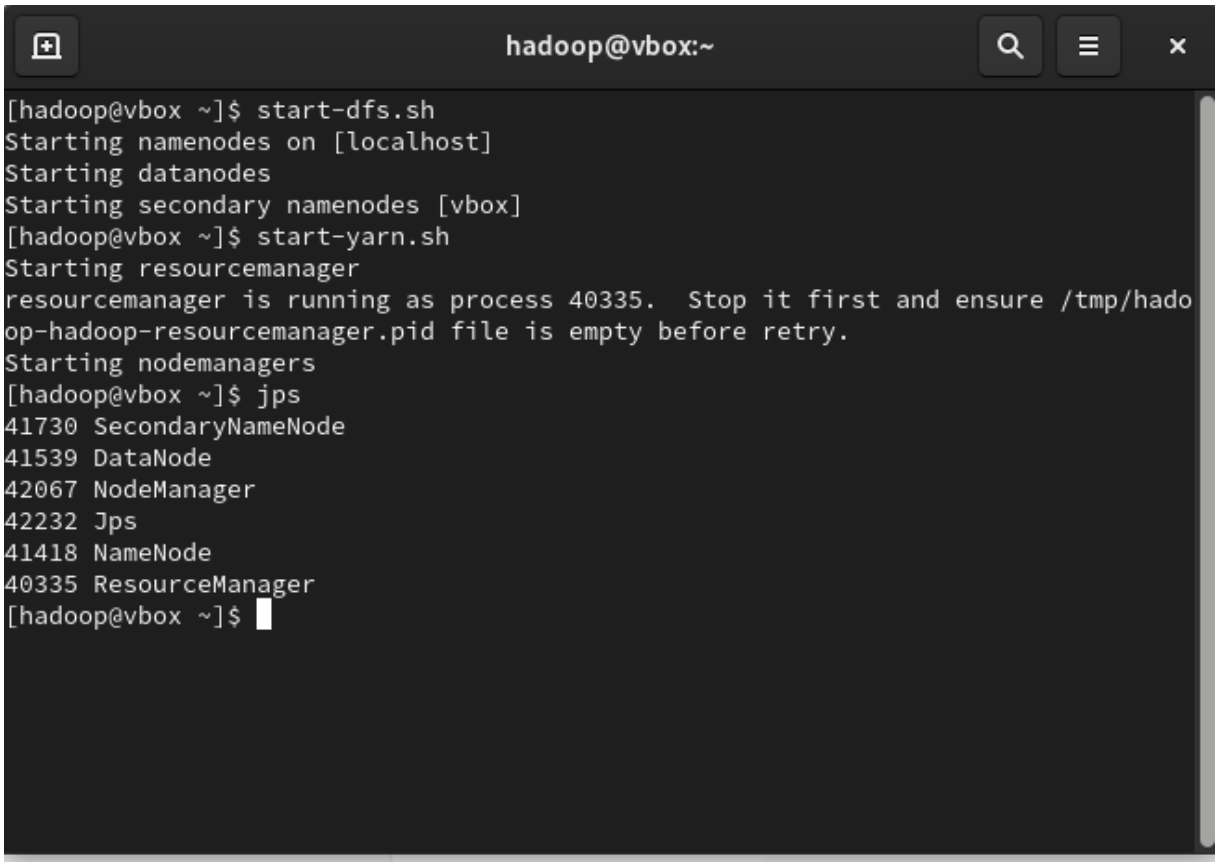


Figura 21. Levantamos servicios de hadoop y yarn

HADOOP

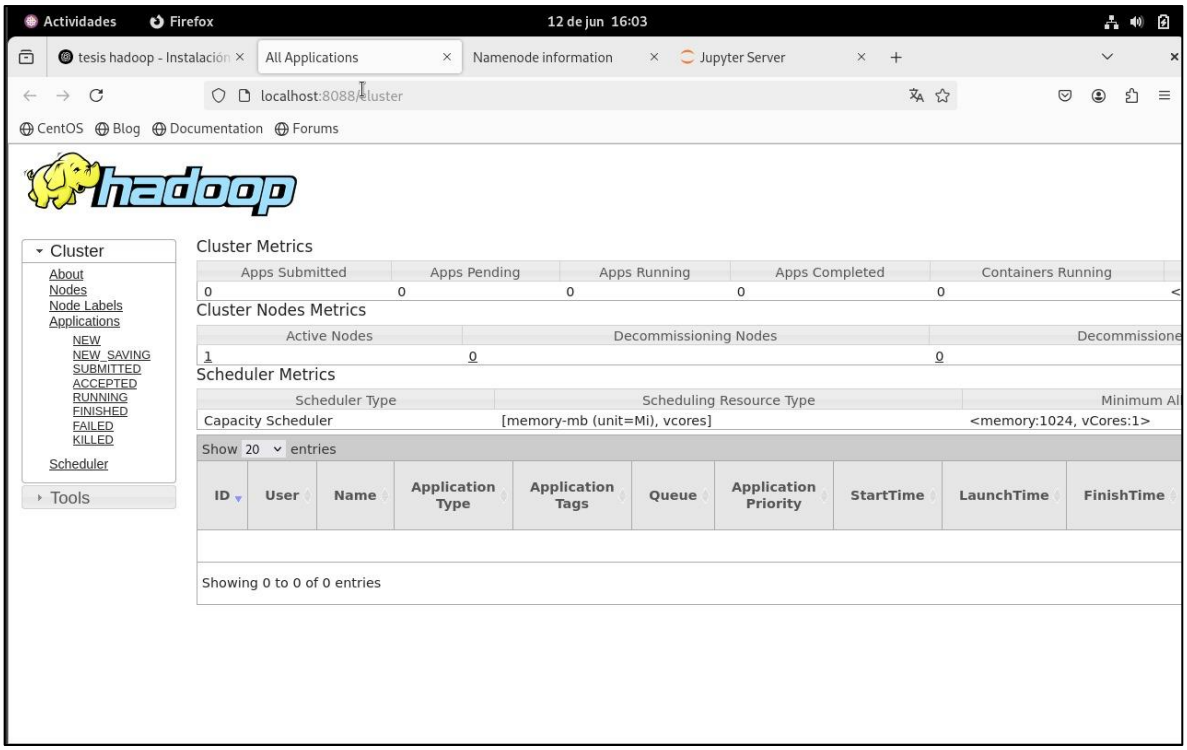


Figura 22. Hadoop

YARN

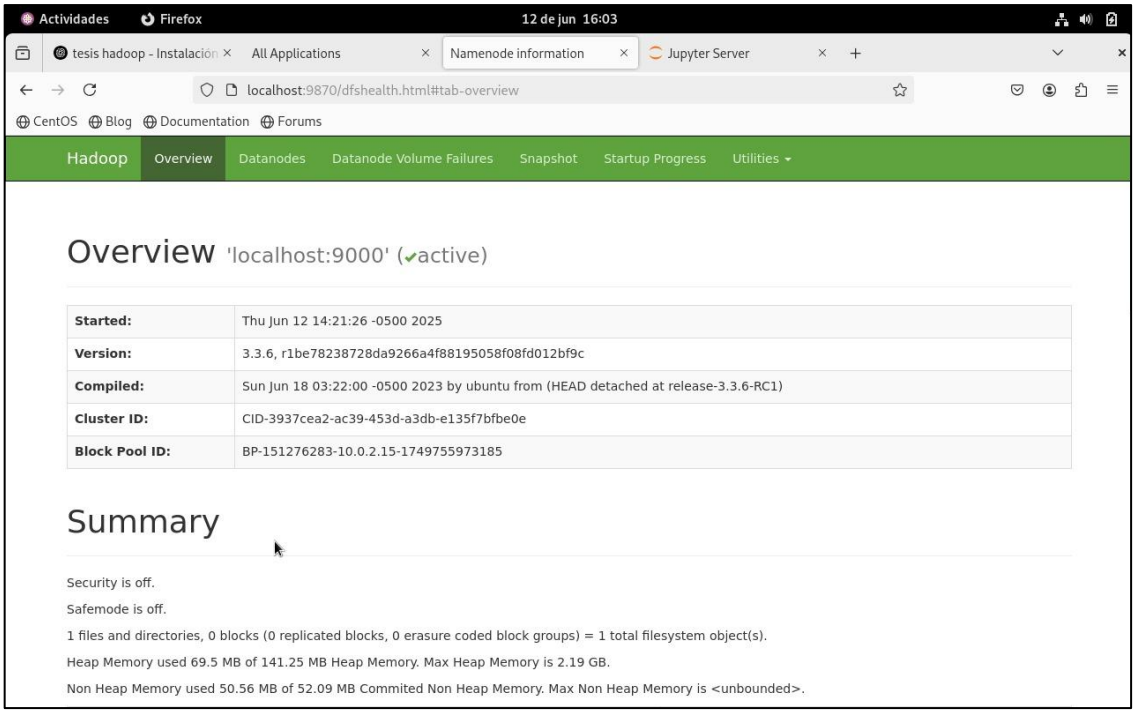


Figura 23. Servicios de yarn

Instalación de spark

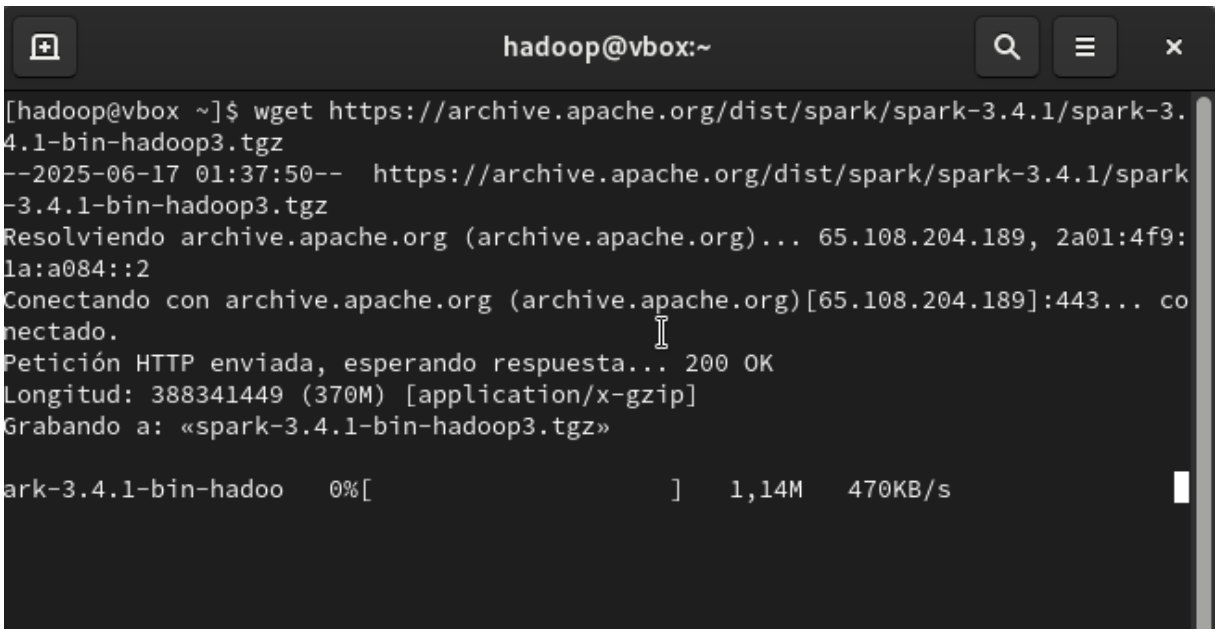


Figura 24. Descarga de apache spark

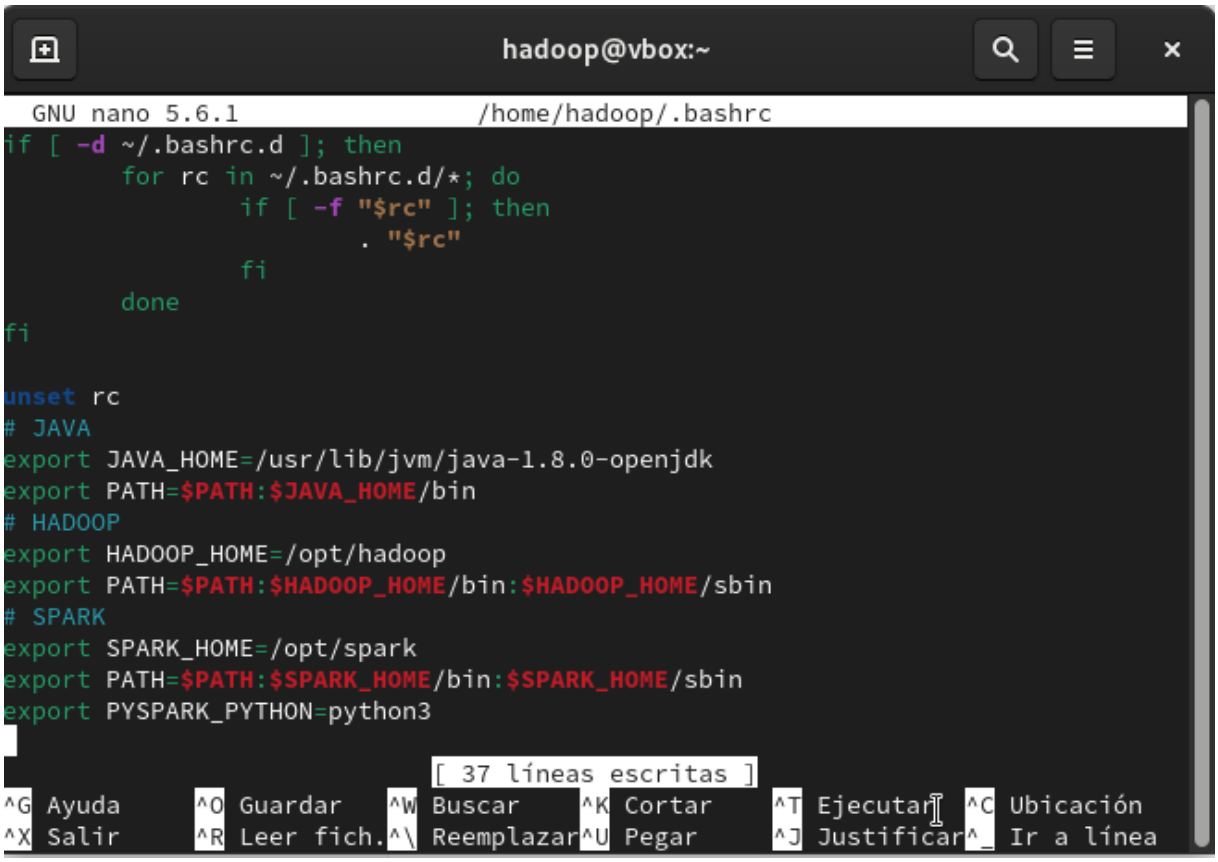


Figura 25. Configuración del bash para spark

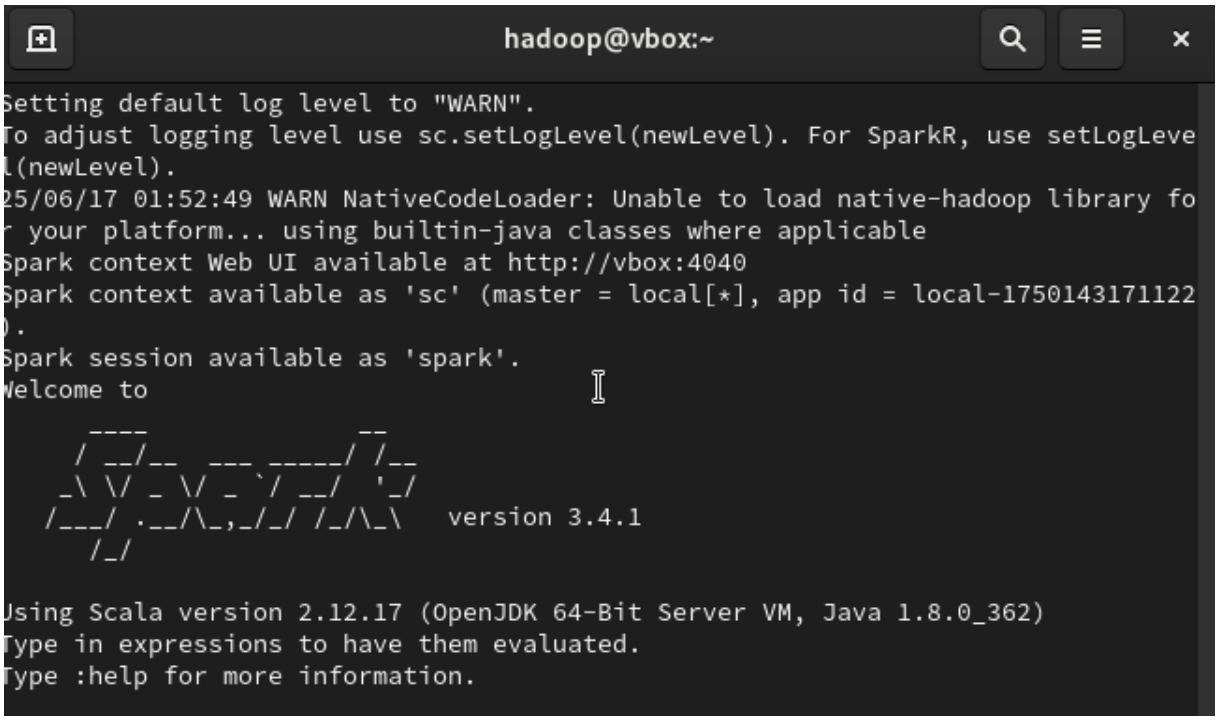


Figura 26. Spark

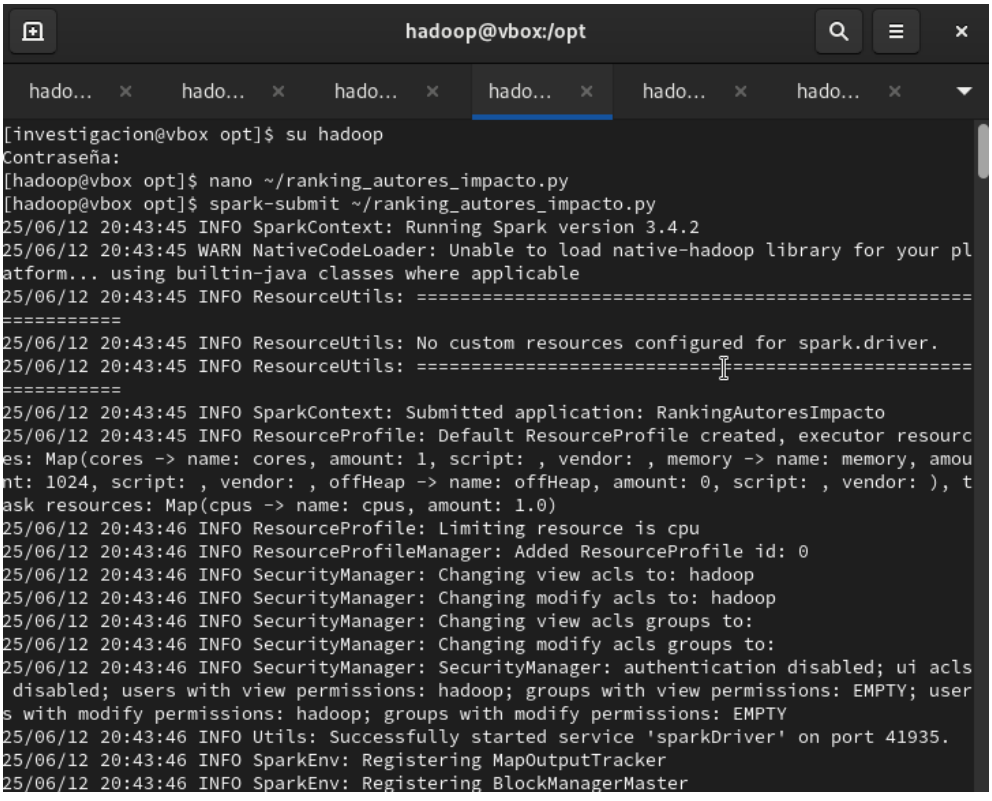


Figura 27. Inicio de pypspark

4.6 PRUEBAS

Las pruebas en un sistema basado en Apache Spark, que se centra en contar cuántas veces aparece un nombre en un conjunto de datos, pueden realizarse mediante un enfoque de pruebas unitarias. Por ejemplo, utilizando unittest en Python, se simula un conjunto de datos con nombres y luego se valida si el sistema cuenta correctamente las repeticiones de cada nombre. El código analiza los datos, agrupándolos por nombre y contando las repeticiones, y luego se comparan los resultados obtenidos con los esperados. Este tipo de pruebas asegura que el sistema funcione correctamente en el conteo de repeticiones, manejando grandes volúmenes de datos de manera eficiente y correcta.

Prueba de pypspark

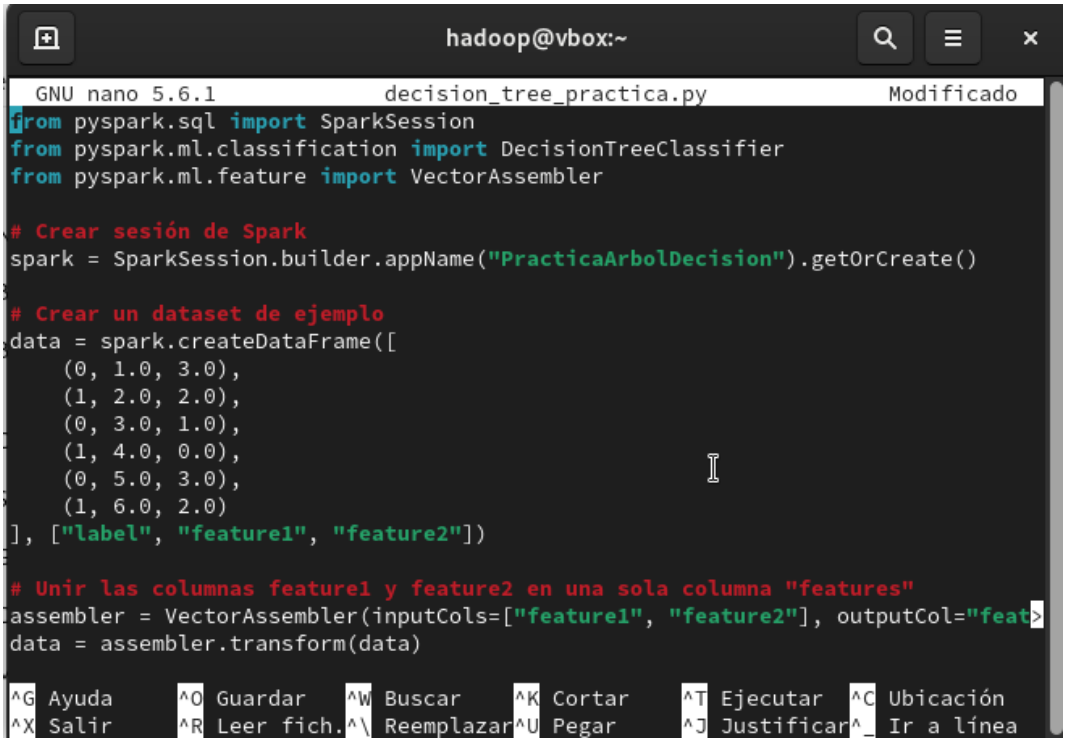


Figura 28. Creación de nano para decisión

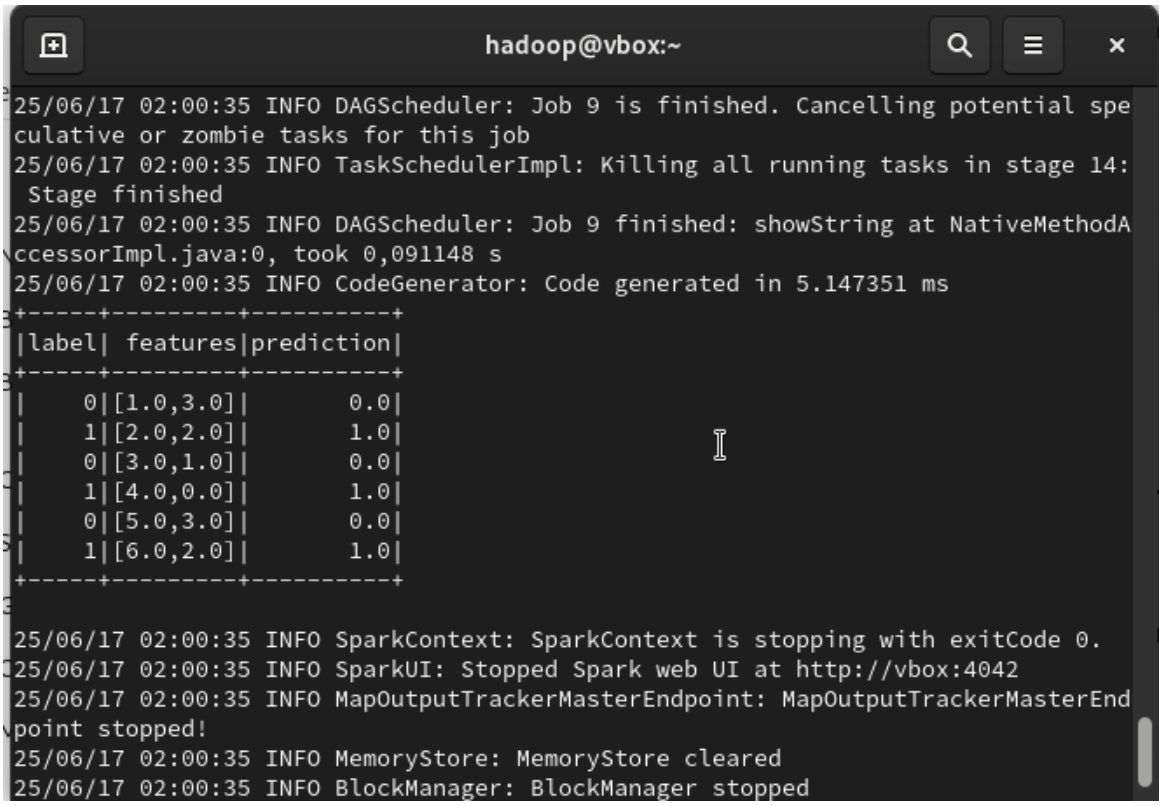


Figura 29. Resultado del árbol de decisiones

Segunda prueba

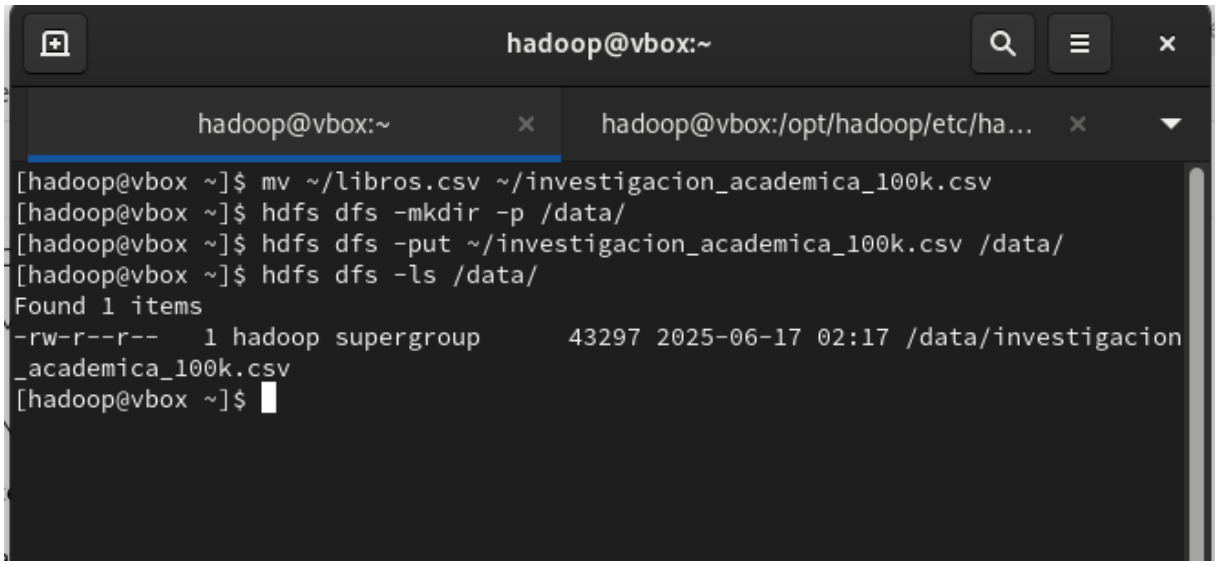


Figura 30. Configuración del database

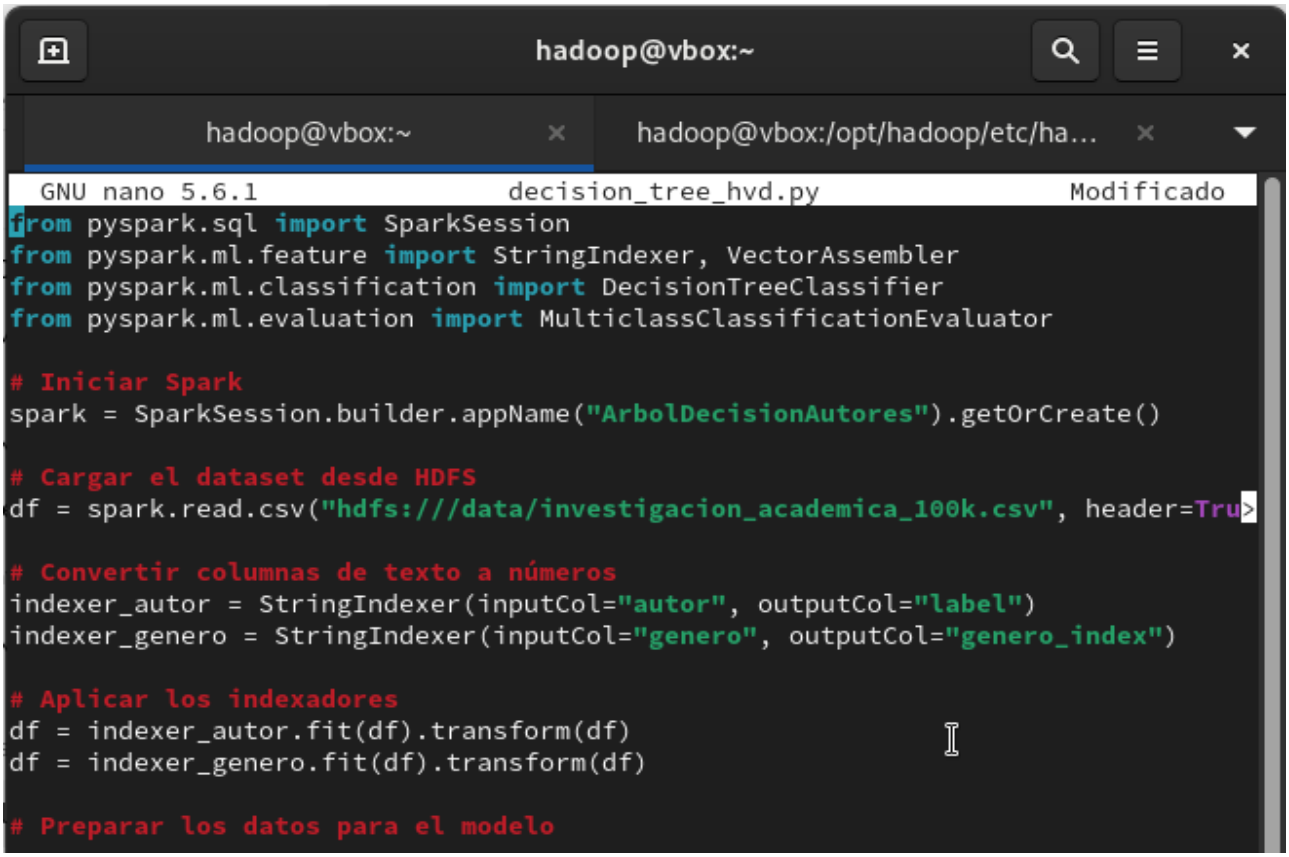


Figura 31. Configuración de un nano para la toma de decisiones

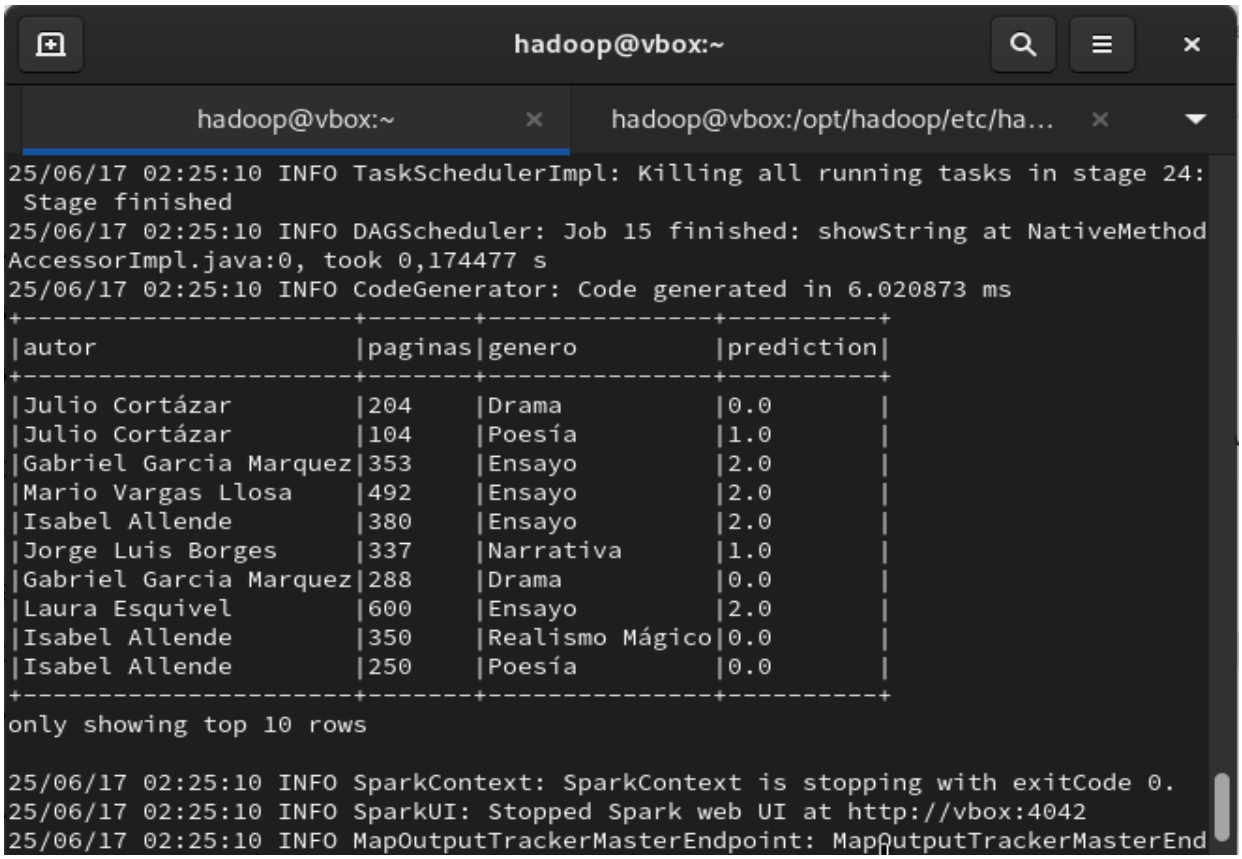


Figura 32. Resultado del ejercicio

Ejercicio de conteo de cantidad de libros

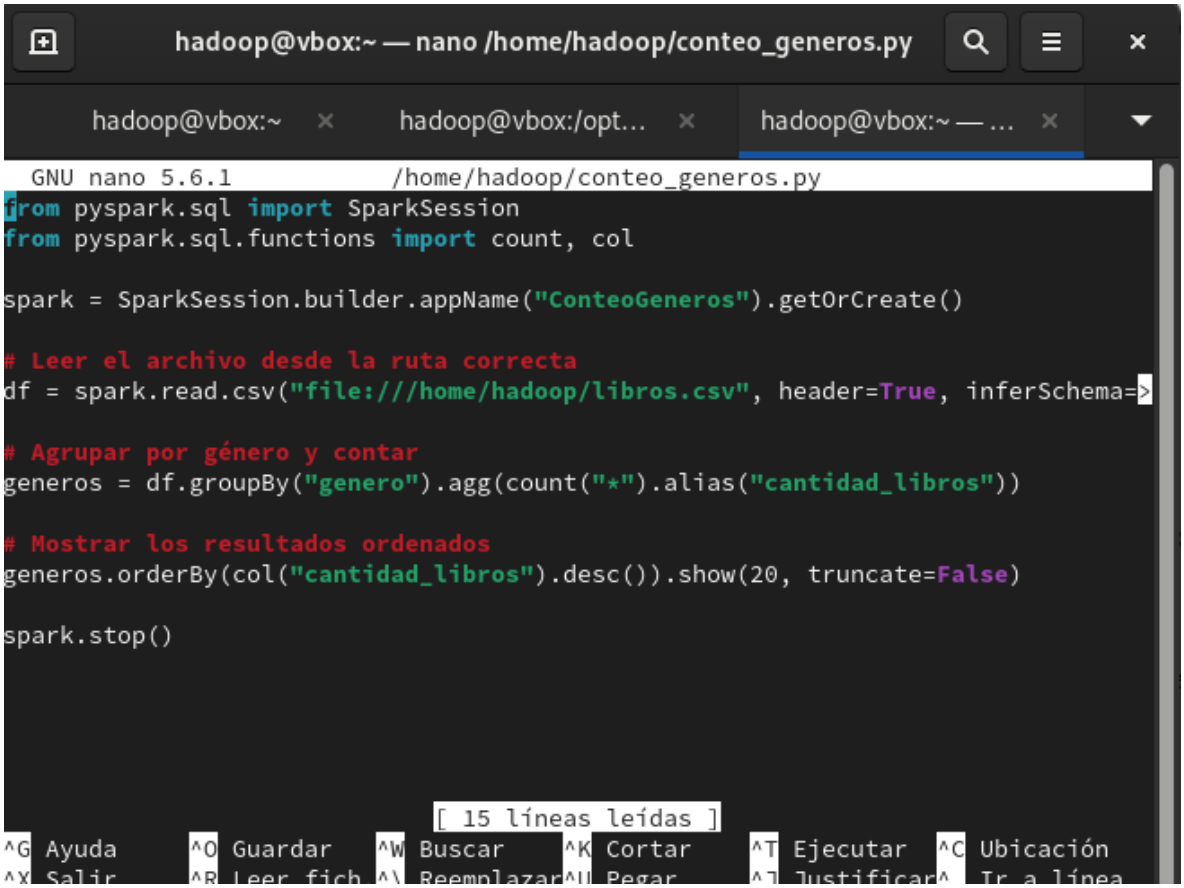


Figura 33. Configuración de un nano conteo_generos.py

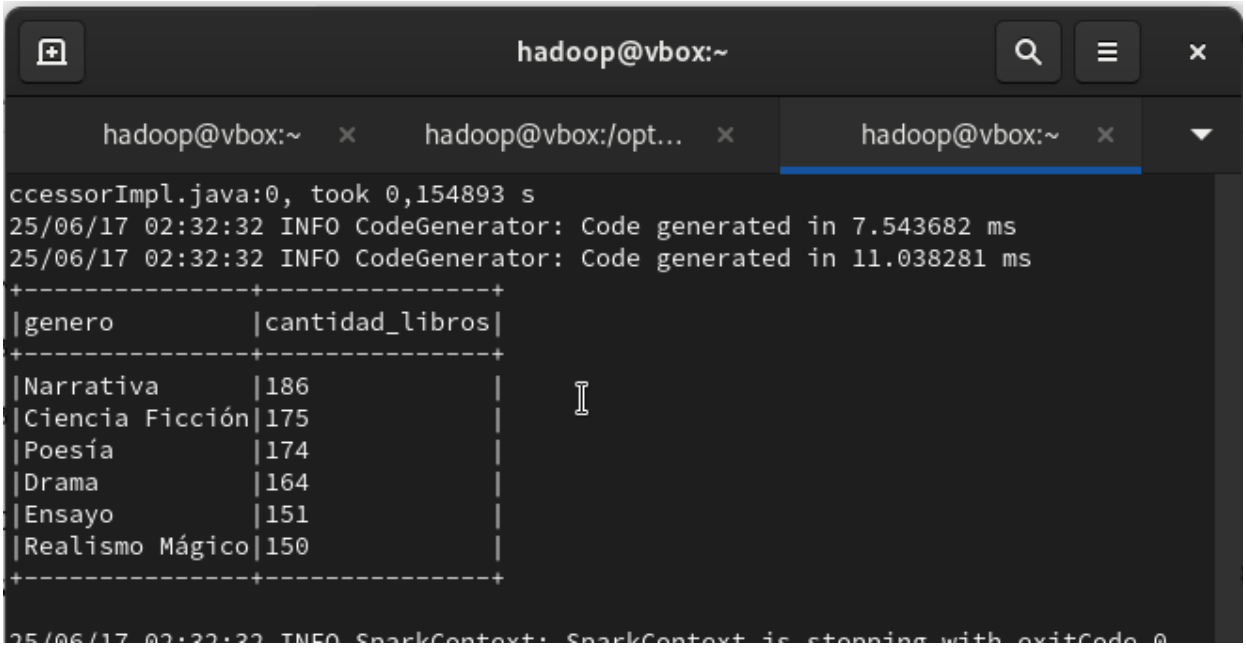


Figura 34. Resultado del ejercicio

Ejercicio de entrenamiento de árbol de decisión

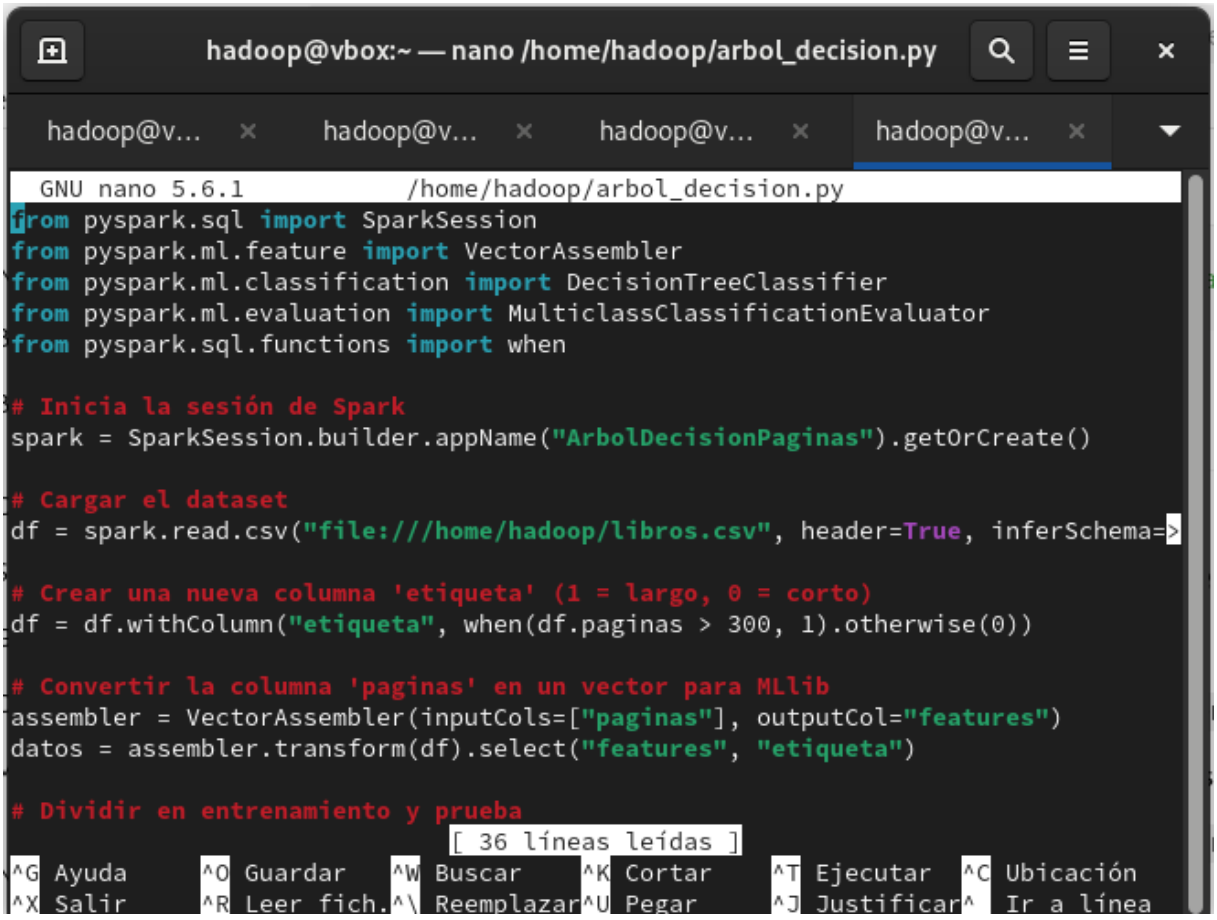


Figura 35. Creación del nano para el árbol decisión

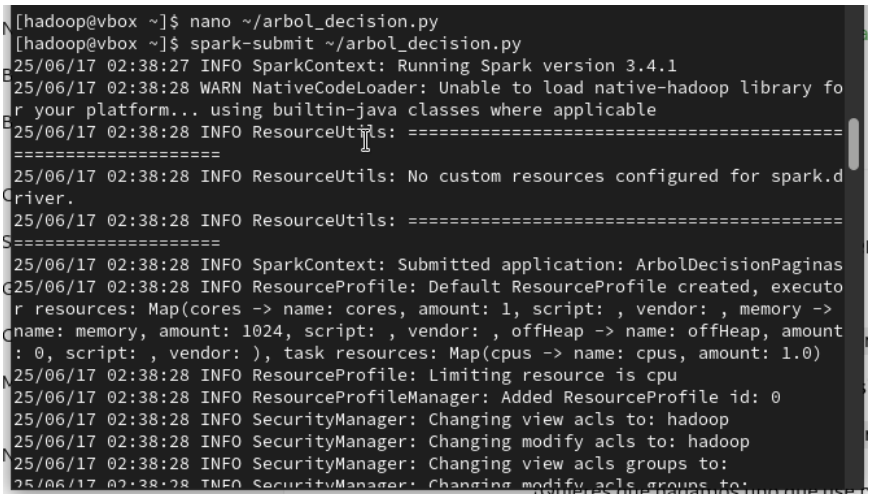


Figura 36. Iniciando entrenamiento

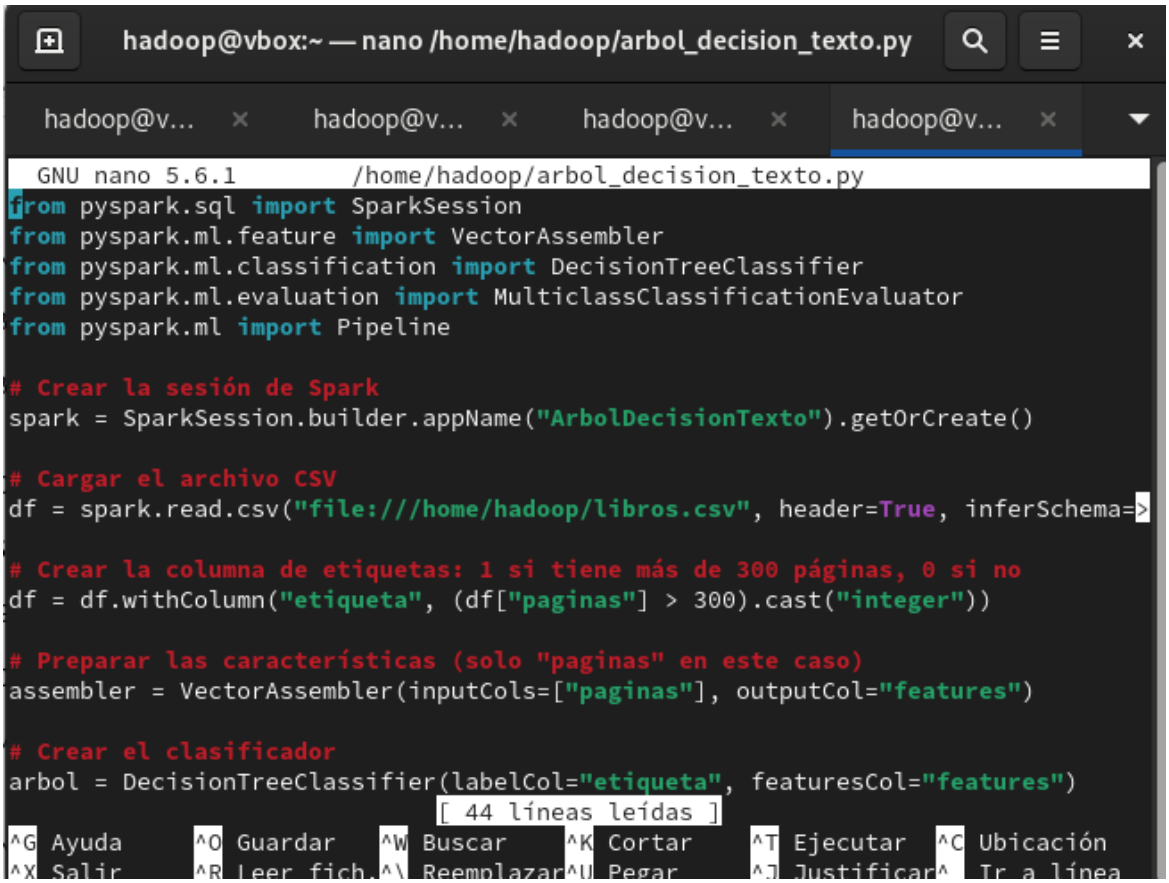


Figura 37. Creación de un nano árbol_decision_texto para mostrar resultados

```
Árbol de decisión entrenado:
DecisionTreeClassificationModel: uid=DecisionTreeClassifier_dc07decdd78c, depth=
1, numNodes=3, numClasses=2, numFeatures=1
  If (feature 0 <= 307.5)
   Predict: 0.0
  Else (feature 0 > 307.5)
   Predict: 1.0
```

Figura 38. Resultado del modelo entrenado

V. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- La información obtenida mediante la fundamentación teórica facilitó la formación de una base firme en los conceptos, herramientas y arquitecturas requeridos para establecer una infraestructura tecnológica enfocada en el procesamiento de algoritmos de Machine Learning en contextos universitarios.
- A través del uso de herramientas de recopilación de datos, se demostró que una gran cantidad de estudiantes percibe como insuficiente la infraestructura tecnológica actual para el aprendizaje práctico de técnicas de aprendizaje automático, lo que evidencia una verdadera necesidad de disponer de recursos especializados y entornos dedicados.
- La propuesta de infraestructura distribuida fundamentada en software libre demostró ser factible para contextos académicos, dado que se consiguió integrar de manera eficaz herramientas como Hadoop, Spark y TensorFlow, posibilitando una implementación eficaz de algoritmos y disminuyendo considerablemente los tiempos de procesamiento.
- El desarrollo del prototipo experimental evidenció que los laboratorios universitarios pueden aprovechar una arquitectura escalable, modular y asequible, que promueva el trabajo en equipo y la experimentación con datos reales, potenciando de esta manera las habilidades de investigación de profesores y alumnos.
- La implementación de la metodología ágil SCRUM en conjunto con el modelo de desarrollo en cascada simplificó la organización de las tareas, la asignación de prioridades y la verificación constante de cada fase del proyecto, garantizando que la implementación se ajuste a las demandas reales del contexto educativo.

5.2. RECOMENDACIONES

- Se aconseja que las autoridades académicas tomen en cuenta la puesta en marcha formal y gradual de la infraestructura sugerida en los laboratorios de computación, iniciando con ensayos controlados y evaluando su influencia en el desempeño académico e investigativo.
- Es crucial seguir formando a profesores y alumnos en el manejo de herramientas avanzadas de aprendizaje automático y computación distribuida, con el objetivo de optimizar el uso de la infraestructura y garantizar la viabilidad del proyecto.
- Se recomienda formar alianzas entre instituciones para intercambiar recursos informáticos, experiencias de implementación y buenas prácticas, lo que potenciará los ecosistemas de innovación en el ámbito académico.
- La infraestructura debe estar respaldada por políticas institucionales explícitas respecto al tratamiento ético de la información, la seguridad de los datos y la salvaguarda de la privacidad, en particular cuando se manejan datasets delicados.
- Para futuros estudios, se recomienda valorar la incorporación de tecnologías emergentes como la inteligencia artificial explicativa, los agentes de colaboración y las plataformas híbridas con recursos en la nube, con la finalidad de mantener la infraestructura en sintonía con las tendencias de la industria.
- Finalmente, se aconseja llevar a cabo una evaluación regular del desempeño del sistema y obtener comentarios de los usuarios, para así poder hacer modificaciones técnicas, expandir la solución a otras capacidades y asegurar su eficacia a largo plazo.

VI. REFERENCIAS BIBLIOGRÁFICAS

Abadi, M. (2019). TensorFlow: Frameworks modernos para machine learning. *IEEE Software Engineering, 15*(2), 120-132.

AIUTE, A. I. (2023). *Estudio sobre la implementacion de infraestructuras tecnologicas en universidades*. https://AIUTE.org

Bergstra, J. (2019). Optimizacion de hiperparámetros en machine learning. *Journal of Artificial Intelligence, 11*(3), 55-70.

Bishop, C. (2019). Aprendizaje supervisado y sus aplicaciones prácticas. *Advances in Machine Learning, 14*(1), 12-24.

Blašković, K., y Čandrlić, S. (2021). Revisión sistemática de metodologías para el desarrollo de sistemas embebidos. *Revista Internacional de Ciencia Avanzada de la Computación y Aplicaciones, 12*(1), 410-425. https://doi.org/https://doi.org/10.14569/IJACSA.2021.0120150

CNCF. (2023). Documentación de Kubernetes. *Fundación para la Computación Nativa en la Nube*. Fundación para la Computación Nativa en la Nube: https://kubernetes.io/docs/

Databricks. (2023). Novedades en PySpark 2023: UDTFs, TorchDistributor, Arrow UDFs. https://doi.org/https://www.databricks.com/blog/2023/12/28/whats-new-pyspark-2023.html

Dünner, C., Parnell, Atasu, K., Sifalakis, M., y Pozidis, H. (2021). Comprendiendo y optimizando el rendimiento de aplicaciones de aprendizaje automático distribuido en Apache Spark.

Fernández, L., y Ríos, M. (2023). Automatización en entornos educativos para el aprendizaje automático. *Revista de Tecnología Educativa., 12*(2), 45-60.

García, F., Corell, A., V, A., y Grande, M. (2022). La transformación digital en la universidad: Infraestructuras y metodologías para la investigación. *Education in the Knowledge Society (EKS)*, 23. https://doi.org/https://doi.org/10.14201/eks.27361

Gómez, R. (2023). Arquitecturas distribuidas para aprendizaje profundo. *Universidad Nacional de Ingeniería*. https://uni.edu.ar/proyectos/gomez2023

Gómez, R. (2023). Big Data en la Educación: Beneficios e Impacto de la Analítica de Datos. *Revista Científica y Tecnológica UPSE, 5*(2), 88-100. https://incyt.upse.edu.ec

González , J., Ramírez , P., y Perez, L. (2023). Impacto del manejo de datos en instituciones educativas. *Revista de Educación superior, 3*, 45-60.

Gonzalez, A., Alba, F., Ordóñez de Pablos, P., y J, G. (2020). Aplicaciones de big data en el ámbito universitario: una revisión sistemática. *Education in the Knowledge Society (EKS), 21*. https://doi.org/https://doi.org/10.14201/eks.23269

González, J., Ramírez, P., y Perez, L. (2023). La competencia digital y desempeño docente en instituciones educativas públicas: Estudio bibliométrico en Scopus. *Revista Científica UISRAEL, 11*(1), 31-45. https://scielo. senecyt.gob.ec

González, M., y Ramírez, J. (2023). *Diseño de infraestructuras académicas modulares para entornos distribuidos de machine learning. Tesis de maestría.* Universidad Nacional de Colombia.

Hernández , P. (2020). Normalizacipon de datos en machine learning. *Data Science Review, 9*(2), 150-160.

IBM. (2023). *La computación en la nube en la educación superior: El papel de IaaS*. Informes de Investigación de IBM: https://www.ibm.com/cloud/

Iqbal, J., Yasin, A., y Omar, M. (2022). Definición de factores de productividad del trabajo en equipo en el desarrollo ágil de software. *Revista Internacional de Ciencia Avanzada, Ingeniería y Tecnología de la Información, 12*(3), 1160-1172. https://doi.org/https://doi.org/10.18517/ijaseit.12.3.13648

Johnson, M. (2019). Redes de alta velocidad y su impacto en marchine learning. *Communication Enfineering, 11*(2), 56-70.

Kohavi, J. (2019). validación cruzada en modelo predictivos. *Advances in Statistical Modeling, 9*(2), 40-55.

Lee , K., Kim, J., y Martínez , C. (2020). Uso de FPGAs para mejorar el rendimiento de algoritmos de IA. *Korean Journal of AI, 5*(2), 67-82.

López, F., Ramírez, G., y Márquez, T. (2021). Plataformas en la nube para la mejora del desempeño académico . *Proceedings of Academic innovations, 8*(2), 120-135.

López, F., Ramírez, G., y Márquez, T. (2021). Soluciones tecnológicas para la educación: Desafíos y oportunidades. *evista de Educación y Tecnología, 8*(2), 120-135. Recuperado de https://www.scielo.edu.uy

Martinez, A. (2022). Procesamiento de datos en clústeres económicos: Un enfoque para inteligencia artificial. *Revista Técnica, 15*(2), 33-45.

Martinez, A. (2022). Transformación digital en las instituciones de educación superior a partir del Covid-19: Madurez tecnológica de los estudiantes en Colombia. *Revista Universidad & Empresa, 23*(2), 71-98.

Matei, A., Popescu, D., y Ionescu, M. (2023). Plataformas de computación distribuida en entornos académicos para aprendizaje automático. *Journal of Advanced Computing Research, 18*(1), 77–89.

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., y Zaharia, M. (2022). MLlib: Aprendizaje automático en Apache Spark. *Journal of Machine Learning Research, 17*(3).

Miller, T. (2021). Virtualización y contenedores en machine learning. *Journal of Computing, 12*(3), 78-90.

Murphy, K. (2020). *Algoritmos no supervisados en big data. Data Mining Journal.*

Ng, A. (2020). Métodos para evitar el sobreajuste en modelos de machine learning. *Artificial Intelligence Journal, 13*(2), 110-125.

Paszke, A. (2019). Pytorch: Una guía para principiantes. *Journal of Reserach in AI, 8*(1), 30-45.

Pressman, R., y Maxim, B. (2021). *Ingeniería del software: Un enfoque práctico (9.ª ed.)*. McGraw-Hill Education.

Project Jupyter. (2023). *Documentación de JupyterHub*. https://jupyter.org/hub

PwC. (2023). *Escalabilidad y arquitectura en inteligencia artificial académica*. Documentos técnicos de PricewaterhouseCoopers: https://www.pwc.com/

Ramírez, P., y Pérez, L. (2022). Impacto de las tecnologías disruptivas en el proceso de enseñanza y aprendizaje en la educación superior. *Revista de Tecnología Educativa, 1*(1), 29-40. https://scielo.senescyt.gob.ec

Ramírez, P., y Pérez, L. (2022). Modelos predictivos en el sector educativo: Un análisis del aprendizaje automátuco. *Journal of Educational Technology , 10*(4), 78-89.

Rocklin, M. (2023). *Dask y Ray para flujos de trabajo escalables en aprendizaje automático*. Actas de la Conferencia de Datos en Python: https://dask.org/

Rodríguez, M., y Sánchez, F. (2022). Prácticas de Extreme Programming para mejorar la calidad del software en proyectos ágiles. *Revista Colombiana de Computación, 23*(1), 71–80.

Ruíz, D. (2021). Infraestructura escalable para el análisis de datos climáticos . *Revista de Ciencias Computacionales, 7*(3), 200-215.

Ruiz, K. (2021). La gestión de la información y el conocimiento a partir de estrategias formativas innovadoras. *Revista de Humanidades y Ciencias Sociales, 4*(1), 124-138.

Schmidt, L. (2021). Tendencias futuras en infraestructura para machine learning. *Future Computing, 14*(1), 25-40.

Schwaber, K., y Sutherland, J. (2020). *La Guía de Scrum: La guía definitiva de Scrum: Las reglas del juego*. Scrum.org: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish.pdf

Serrador, P., y Pinto, J. (2025). ¿Funciona Ágil? Un análisis cuantitativo del éxito de proyectos ágiles. *Revista Internacional de Gestión de Proyectos, 33*(5), 1040–1051. https://doi.org/https://doi.org/10.1016/j.ijproman.2015.01.006

Smith, J. (2020). Optimización de infraestructura tecnológica en machine learning. *Tech Journal, 8*(1), 20-35.

Sommerville, I. (2020). Ingeniería del software (10.ª ed.). *Pearson Educación*.

Soto, M., Mendoza, D., y Romero, C. (2023). Aplicación de tecnologías Big Data en plataformas de gestión educativa. *Revista Iberoamericana de Educación Digital, 14*(1), 55-68.

Sutton , R. (2019). Aprendizaje por refuerzo: Teoría y práctica. *Machine Learning Research, 10*(1), 80-95.

Tang, S., He, B., Yu, C., Li, Y., y Li, K. (2020). Una revisión del ecosistema Spark para el procesamiento de Big Data. *Prepublicación en arXiv*. https://doi.org/https://arxiv.org/abs/1811.04815

TechTarget. (2023). *Top 10 infrastructure trends to watch in 2024. Recuperado de*. TechTarget: https://www.techtarget.com/searchdatacenter/feature/Top-10-infrastructure-trends-to-watch-in-2024

UNESCO. (2023). *Guía sobre ética de la inteligencia artificial en educación: privacidad, transparencia y protección de datos*. UNESCO.

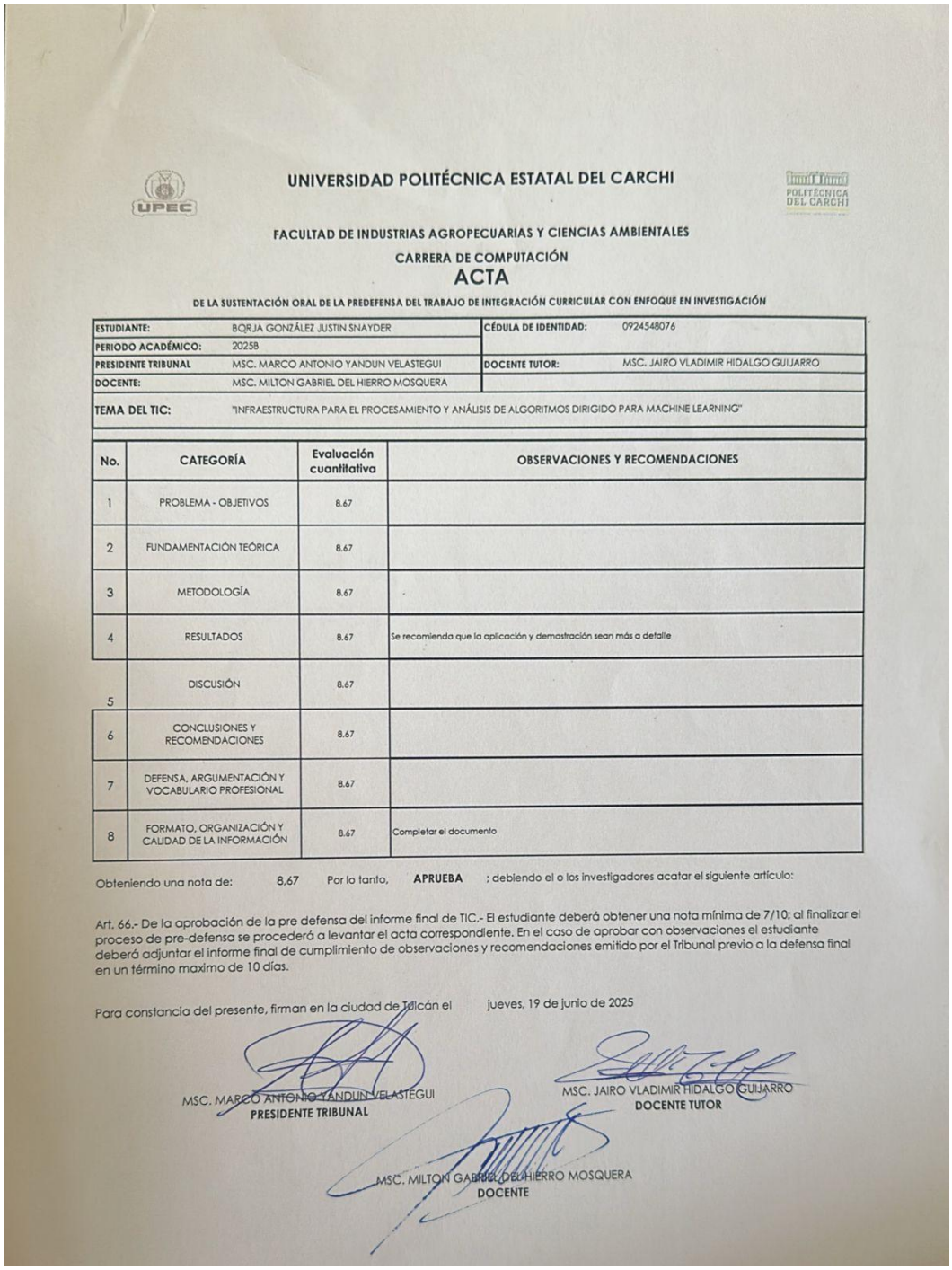
White, T. (2022). *Hadoop: La guía definitiva (4.ª ed.)*. O'Reilly Media.

Zaharia, M. (2020). Hadoop y Spark: Plataformas para Big Data. *Computing Advances, 12*(2), 90-105.

Zhao, Y., y Liu, H. (2023). Análisis comparativo de TensorFlow y PyTorch en la investigación académica. *Revista Internacional de Aprendizaje Profundo, 11*(3), 133-150.

VII. ANEXOS

Anexo 1. Acta de la sustentación de Pre-defensa del TIC



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI

FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

ACTA

DE LA SUSTENTACIÓN ORAL DE LA PREDEFENSA DEL TRABAJO DE INTEGRACIÓN CURRICULAR CON ENFOQUE EN INVESTIGACIÓN

ESTUDIANTE:	BORJA GONZÁLEZ JUSTIN SNAYDER	**CÉDULA DE IDENTIDAD:**	0924548076
PERIODO ACADÉMICO:	2025B		
PRESIDENTE TRIBUNAL	MSC. MARCO ANTONIO YANDUN VELASTEGUI	**DOCENTE TUTOR:**	MSC. JAIRO VLADIMIR HIDALGO GUIJARRO
DOCENTE:	MSC. MILTON GABRIEL DEL HIERRO MOSQUERA		
TEMA DEL TIC:	"INFRAESTRUCTURA PARA EL PROCESAMIENTO Y ANÁLISIS DE ALGORITMOS DIRIGIDO PARA MACHINE LEARNING"		

No.	CATEGORÍA	Evaluación cuantitativa	OBSERVACIONES Y RECOMENDACIONES
1	PROBLEMA - OBJETIVOS	8.67	
2	FUNDAMENTACIÓN TEÓRICA	8.67	
3	METODOLOGÍA	8.67	
4	RESULTADOS	8.67	Se recomienda que la aplicación y demostración sean más a detalle
5	DISCUSIÓN	8.67	
6	CONCLUSIONES Y RECOMENDACIONES	8.67	
7	DEFENSA, ARGUMENTACIÓN Y VOCABULARIO PROFESIONAL	8.67	
8	FORMATO, ORGANIZACIÓN Y CALIDAD DE LA INFORMACIÓN	8.67	Completar el documento

Obteniendo una nota de: 8,67 Por lo tanto, **APRUEBA** ; debiendo el o los investigadores acatar el siguiente artículo:

Art. 66.- De la aprobación de la pre defensa del informe final de TIC.- El estudiante deberá obtener una nota mínima de 7/10; al finalizar el proceso de pre-defensa se procederá a levantar el acta correspondiente. En el caso de aprobar con observaciones el estudiante deberá adjuntar el informe final de cumplimiento de observaciones y recomendaciones emitido por el Tribunal previo a la defensa final en un término maximo de 10 días.

Para constancia del presente, firman en la ciudad de Tulcán el jueves, 19 de junio de 2025

MSC. MARCO ANTONIO YANDUN VELASTEGUI
PRESIDENTE TRIBUNAL

MSC. JAIRO VLADIMIR HIDALGO GUIJARRO
DOCENTE TUTOR

MSC. MILTON GABRIEL DEL HIERRO MOSQUERA
DOCENTE

Anexo 2. Rubrica de la sustentación de Pre-defensa del TIC

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI

FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

RÚBRICA DE EVALUACIÓN DE LA SUSTENTACIÓN ORAL DE LA PREDEFENSA DEL TRABAJO DE INTEGRACIÓN CURRICULAR CON ENFOQUE EN INVESTIGACIÓN

ESTUDIANTE:	BORJA GONZALEZ JUSTIN SNAYDER	CÉDULA DE IDENTIDAD:	0924548076
PERIODO ACADÉMICO:	2025A	FECHA: 19 de junio de 2025	HORA: 15:00
PRESIDENTE TRIBUNAL:	MSC. MARCO ANTONIO YANDUN VELASTEGUI	DOCENTE TUTOR: MSC. JAIRO VLADIMIR HIDALGO GUIJARRO	
DOCENTE:	MSC. MILTON GABRIEL DEL HIERRO MOSQUERA	AULA: EDIFICIO DE AULAS:	
TEMA DEL TIC:	"INFRAESTRUCTURA PARA EL PROCESAMIENTO Y ANÁLISIS EN ALGORITMOS DIRIGIDO PARA MACHINE LEARNING"		

	No.	CATEGORÍA	CRITERIO ÓPTIMO DE EVALUACIÓN	PRESIDENTE TRIBUNAL	DOCENTE TUTOR	DOCENTE
SUSTENTACIÓN ORAL DEFENSA	1	PROBLEMA - OBJETIVOS	Se expone el planteamiento, formulación y justificación, los objetivos son expuestos como sistémicos para alcanzar el objetivo general; las preguntas de investigación aportan a entender la que se quiere investigar y son coherentes con los objetivos.	9	8.5	8.5
	2	FUNDAMENTACIÓN TEÓRICA	Es un marco de referencia para el desarrollo e interpretación de los resultados de la investigación. Los antecedentes investigativos incluidos tienen relación con el tema planteado.	9	8.5	8.5
	3	METODOLOGÍA	El estudiante explicó el enfoque de la investigación de manera lógica al análisis estadístico, la población, muestra, técnicas e instrumentos presentados, permitiendo entender que el informe es consistente en resultados y discusión.	9	8.5	8.5
	4	RESULTADOS	Se analizó la relación entre las variables de manera cualitativa, cuantitativa y fueron representativas a la profesión. Expuso gráficos, figuras, tablas de frecuencia y contingencia coherentes y de acuerdo a la metodología de investigación. Los datos fueron presentados de forma clara y efectiva a lo observado y no exigen interpretaciones.	9	8.5	8.5
	5	DISCUSIÓN	La discusión expuesta y defendida establece la relación de los objetivos propuestos, con los antecedentes de la investigación y el tema.	9	8.5	8.5
	6	CONCLUSIONES Y RECOMENDACIONES	Las conclusiones y recomendaciones expuestas, son claras, concisas y acordes a los objetivos y resultados de la investigación.	9	8.5	8.5
	7	DEFENSA, ARGUMENTACIÓN Y VOCABULARIO PROFESIONAL	El estudiante demostró conocimiento y seguridad del objeto de estudio. Relacionó conceptos y teorías. El vocabulario utilizado fue acorde a la terminología de la profesión con un volumen de voz adecuado. Hizo un uso correcto del tiempo. Utilizo recursos didácticos apropiados.	9	8.5	8.5
			PROMEDIO SOBRE SIETE		6.07	
DOCUMENTO ESCRITO	8	FORMATO, ORGANIZACIÓN Y CALIDAD DE LA INFORMACIÓN	El formato, la organización de contenidos, redacción, uso de gramática y ortografía, aplicación de normas de citas y referencias cumplen con el formato de la UPEC.	9	8.5	8.5
			PROMEDIO SOBRE TRES		2.60	
					8,67	

Art. 66.- De la aprobación de la pre defensa del informe final de TIC.- El estudiante deberá obtener una nota mínima de 7/10; al finalizar el proceso de pre-defensa se procederá a levantar el acta correspondiente. En el caso de aprobar con observaciones el estudiante deberá adjuntar el informe final de cumplimiento de observaciones y recomendaciones emitido por el Tribunal previo a la defensa final en un término máximo de 10 días.

Art. 67.- De la no aprobación de la pre defensa del TIC.- Si el estudiante no aprueba la pre defensa tendrá un término de 30 días para realizar los cambios y presentarse por una sola ocasión a una segunda pre defensa, para ello entregará la solicitud dirigida a la Dirección de Carrera. Si el estudiante en la pre defensa adicional no alcanzara la nota establecida para su aprobación, deberá solicitar por una sola ocasión el cambio de opción de titulación.

MSC. MARCO ANTONIO YANDUN VELASTEGUI
PRESIDENTE TRIBUNAL

MSC. JAIRO VLADIMIR HIDALGO GUIJARRO
DOCENTE TUTOR

MSC. MILTON GABRIEL DEL HIERRO MOSQUERA
DOCENTE

Anexo 3. Certificado del abstract por parte de idiomas

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI FOREIGN AND NATIVE LANGUAGES CENTER

ABSTRACT- EVALUATION SHEET

NAME: Justin Snayder Borja González
DATE: Miércoles , 27 de agosto de 2025
Topic: "Infraestructura para el procesamiento y análisis en algoritmos dirigido para Machine Learning"
MARKS AWARDED **QUANTITATIVE AND QUALITATIVE**

VOCABULARY AND WORD USE	Use new learnt vocabulary and precise words related to the topic	Use a little new vocabulary and some appropriate words related to the topic	Use basic vocabulary and simplistic words related to the topic	Limited vocabulary and inadequate words related to the topic
	EXCELLENT: 2 ☐	GOOD: 1,5 ☐	AVERAGE: 1 ☐	LIMITED: 0,5 ☐
WRITING COHESION	Clear and logical progression of ideas and supporting paragraphs.	Adequate progression of ideas and supporting paragraphs.	Some progression of ideas and supporting paragraphs.	Inadequate ideas and supporting paragraphs.
	EXCELLENT: 2 ☐	GOOD: 1,5 ☐	AVERAGE: 1 ☐	LIMITED: 0,5 ☐
ARGUMENT	The message has been communicated very well and identify the type of text	The message has been communicated appropriately and identify the type of text	Some of the message has been communicated and the type of text is little confusing	The message hasn't been communicated and the type of text is inadequate
	EXCELLENT: 2 ☐	GOOD: 1,5 ☐	AVERAGE: 1 ☐	LIMITED: 0,5 ☐
CREATIVITY	Outstanding flow of ideas and events	Good flow of ideas and events ■	Average flow of ideas and events	Poor flow of ideas and events
	EXCELLENT: 2 ☐	GOOD: 1,5	AVERAGE: 1 ☐	LIMITED: 0,5 ☐
SCIENTIFIC SUSTAINABILITY	Reasonable, specific and supportable opinion or thesis statement	Minor errors when supporting the thesis statement	Some errors when supporting the thesis statement	Lots of errors when supporting the thesis statement
	EXCELLENT: 2 ■	GOOD: 1,5 ☐	AVERAGE: 1 ☐	LIMITED: 0,5 ☐

TOTAL/AVERAGE	9 - 10: EXCELLENT 7 - 8,9: GOOD 5 - 6,9: AVERAGE 0 - 4,9: LIMITED	TOTAL 9

1

Anexo 4. Encuestas dirigidas a estudiantes

Infraestructura para el procesamiento y análisis de algoritmos dirigidos maching learning

1. En su universidad existe una infraestuctura basica para el procesamiento y análisis en algoritmos dirigido para machine learnig *

- ◯ Si
- ◯ No

2. Los recursos tecnológicos actuales que tienen le permiten aprender *machine learning* de forma efectiva *

- ◯ totalmente en desacuerdo
- ◯ en desacuerdo
- ◯ Ni de acuerdo ni en desacuerdo
- ◯ de acuerdo
- ◯ totalmente de acuerdo

3. Es importante tener acceso a herramientas específicas para practicar y desarrollar proyectos de *machine learning*. *

- ◯ totalmente en desacuerdo
- ◯ en desacuerdo
- ◯ Ni de acuerdo ni en desacuerdo
- ◯ de acuerdo
- ◯ totalmente de acuerdo

4. El aprendizaje de *machine learning* está limitado por la falta de infraestructura adecuada, como computadoras potentes o programas especializados. *

- ◯ totalmente en desacuerdo
- ◯ en desacuerdo
- ◯ Ni de acuerdo ni en desacuerdo
- ◯ de acuerdo
- ◯ totalmente de acuerdo

5. Los recursos gratuitos disponibles para aprender *machine learning* son suficientes para empezar a practicar. *

- ◯ totalmente en desacuerdo
- ◯ en desacuerdo
- ◯ Ni de acuerdo ni en desacuerdo
- ◯ de acuerdo
- ◯ totalmente de acuerdo

6. Consideras que las computadoras que utilizas suelen ser rápidas y eficientes para trabajar con algoritmos básicos de *machine learning*. *

- ◯ totalmente en desacuerdo
- ◯ En desacuerdo
- ◯ Ni de acuerdo ni en desacuerdo
- ◯ De acuerdo
- ◯ totalmente de acuerdo

7. Te gustaría tener acceso a recursos más especializados para realizar proyectos más avanzados de *machine learning*. *

- ◯ totalmente en desacuerdo
- ◯ En desacuerdo
- ◯ Ni de acuerdo ni en desacuerdo
- ◯ De acuerdo
- ◯ totalmente de acuerdo

8. Crees que contar con mejores herramientas tecnológicas te ayudaría a entender y aplicar conceptos de *machine learning* más fácilmente. *

- ◯ totalmente en desacuerdo
- ◯ En desacuerdo
- ◯ Ni de acuerdo ni en desacuerdo
- ◯ De acuerdo
- ◯ totalmente de acuerdo

9. Considero que sería útil contar con un espacio dedicado para practicar proyectos de *machine learning*. *

- ◯ totalmente en desacuerdo
- ◯ En desacuerdo
- ◯ Ni de acuerdo ni en desacuerdo
- ◯ De acuerdo
- ◯ totalmente de acuerdo

10. Encontrar herramientas adecuadas para *machine learning* es un desafío constante en mi proceso de aprendizaje. *

- ◯ totalmente en desacuerdo
- ◯ En desacuerdo
- ◯ Ni de acuerdo ni en desacuerdo
- ◯ De acuerdo
- ◯ totalmente de acuerdo

Anexo 5. Manual de usuario

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI

FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES

CARRERA DE COMPUTACIÓN

MANUAL DE USUARIO DE LA INFRAESTRUCTURA PARA EL PROCESAMIENTO Y ANÁLISIS DE ALGORITMOS DIRIGIDOS PARA MACHINE LEARNING

Infraestructura distribuida para el análisis académico con Machine Learning

Junio 2025

Introducción

1.1 Propósito del Manual

Este manual describe el uso, instalación y validación del prototipo de infraestructura distribuida desarrollado en una laptop personal con máquinas virtuales, diseñado para ejecutar algoritmos de Machine Learning usando herramientas de código abierto.

1.2 Alcance

Este documento está dirigido a estudiantes e investigadores que deseen replicar el entorno técnico para experimentación académica. El sistema fue validado en un entorno simulado compuesto por un nodo maestro y un nodo esclavo.

2. Componentes del Sistema

Componente	**Descripción**
Nodo maestro	Configurado con Hadoop, Spark y Jupyter
Nodo esclavo	Nodo de apoyo para tareas distribuidas
Virtualización	Oracle VirtualBox con Ubuntu Server
Lenguajes y herramientas	PySpark, Hadoop, Spark

3. Requisitos del Sistema

Hardware mínimo para pruebas locales

- Procesador: Intel i5 o equivalente
- RAM: 8 GB
- Almacenamiento: 100 GB libre
- Red: Conexión virtual NAT o puente

Software

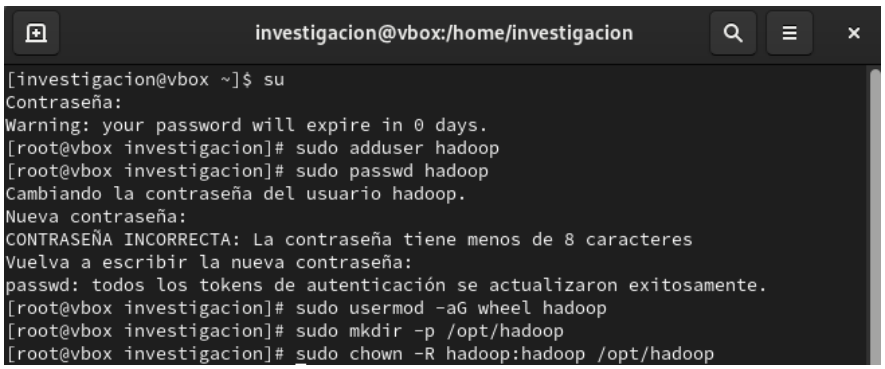
- CentOS 9 en cada VM
- Hadoop 3.3.6
- Spark 3.5.0

- Python 3.10
- OpenSSH, Java 8

4. Instalación de los Servicios1

4.1 Instalación básica del entorno

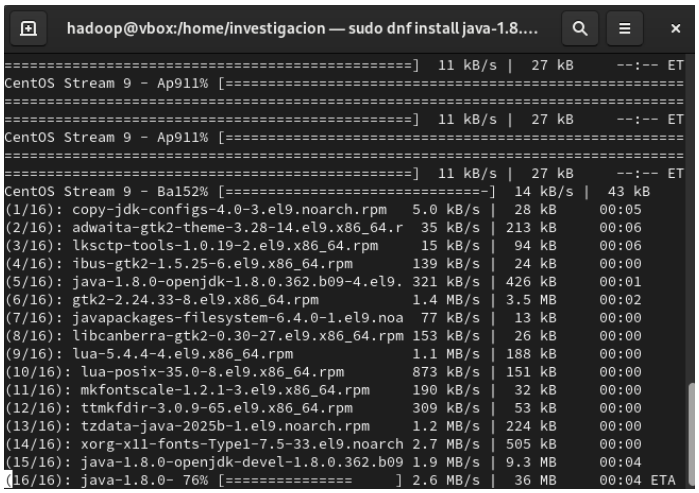
En este paso nosotros creamos un usuario llamado "hadoop" el cual nos servirá para poder trabajar en cada una de las instalaciones que nosotros necesitemos



- [investigacion@vbox ~]$ su
- Contraseña:
- Warning: your password will expire in 0 days.
- [root@vbox investigacion]# sudo adduser hadoop
- [root@vbox investigacion]# sudo passwd hadoop
- Cambiando la contraseña del usuario hadoop.
- passwd: todos los tokens de autenticación se actualizaron exitosamente.
- [root@vbox investigacion]# sudo usermod -aG Wheel hadoop
- [root@vbox investigacion]# sudo mkdir -p /opt/hadoop
- [root@vbox investigacion]# Sudo chown -R hadoop:hadoop /opt/hadoop

4.2 Configuración de java JDK 8 y Hadoop

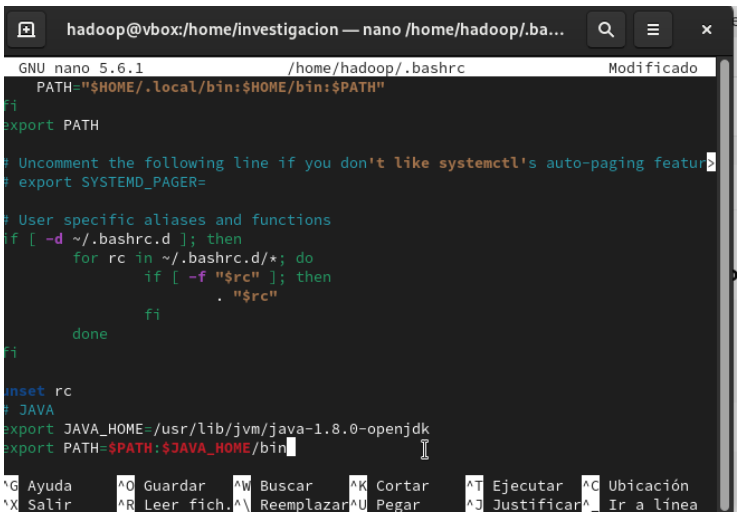
Lo principal para toda la configuración debemos tener instalado lo que es el Java JDK 8 debido a que es necesario para el buen funcionamiento de hadoop y sus servicios debido a que hadoop está basado en lenguaje java y sin el JDK e sistema no puede implementar ni ejecutar ningún código referente a hadoop



- sudo apt update
- sudo apt install openjdk-8-jdk -y

4.3 Configuración nano ~/bash

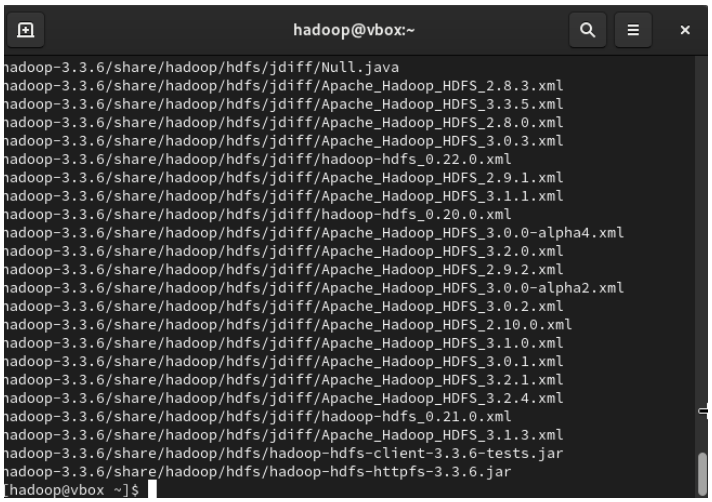
Es necesario configurar la variable de entorno JAVA_HOME porque Hadoop necesita saber en qué ubicación está instalado Java para poder ejecutarse correctamente. Como Hadoop está construido en lenguaje Java, requiere acceder al compilador y al entorno de ejecución de Java que proporciona el **JDK 8**.



- export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
- export PATH=$PATH:$JAVA_HOME/bin

Descarga de Hadoop

Es importante descargar Hadoop porque es el software base que permite implementar una infraestructura para el procesamiento distribuido de datos y poder utilizar los servicios que ofrece como son los HDFS, yarn, spark.



- wget https://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz
- tar -xvzf hadoop-3.3.6.tar.gz
- sudo mv hadoop-3.3.6 /opt/Hadoop
- ls /opt/Hadoop

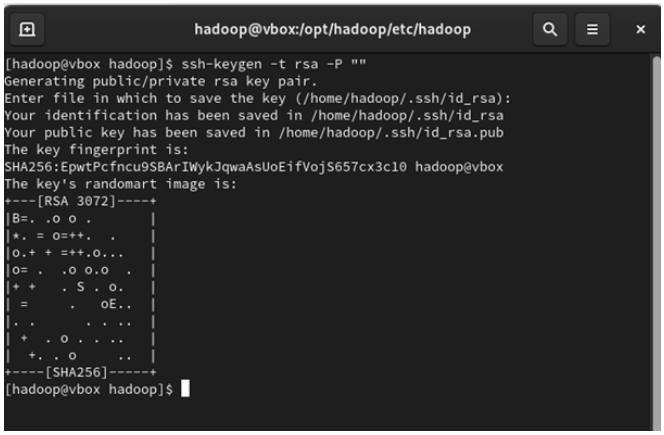
configuración del bash

La configuración mostrada en el archivo core-site.xml es esencial porque define el sistema de archivos por defecto que usará Hadoop. Al establecer fs.defaultFS con el valor hdfs://localhost:9000, se indica que el sistema trabajará con HDFS y que el NameNode estará disponible en el puerto 9000 de la máquina local.

- <property>
  <name>fs.defaultFS</name>
  <value>hdfs://localhost:9000</value>
  </property>

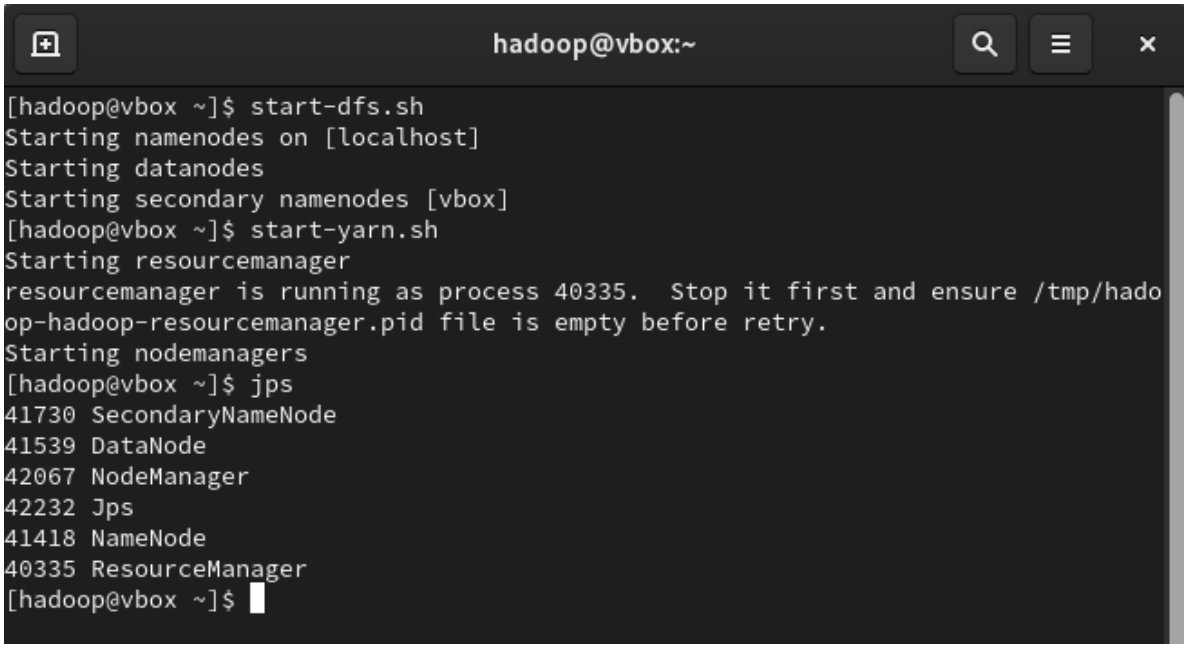
Configuracion del ssh

La imagen muestra el proceso de generación de una clave SSH con el comando ssh-keygen -t rsa -P '', lo cual es un paso fundamental para permitir la conexión automática entre el nodo maestro y los nodos esclavos en Hadoop sin necesidad de ingresar contraseña



Levantamiento de servicios

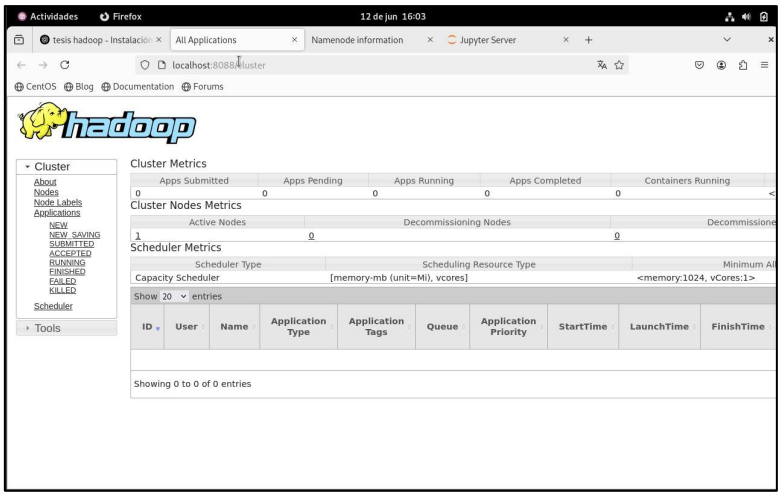
Esta imagen muestra el proceso de inicio de los servicios principales de Hadoop mediante los comandos start-dfs.sh y start-yarn.sh. Estos comandos levantan los demonios esenciales del sistema distribuido: el NameNode, DataNode, SecondaryNameNode, ResourceManager y NodeManager. El comando jps se utiliza para verificar que estos servicios estén corriendo correctamente. Esta verificación es crucial porque confirma que el clúster está activo y listo para procesar tareas distribuidas



- start-dfs.sh
- start-yarn.sh

hadoop

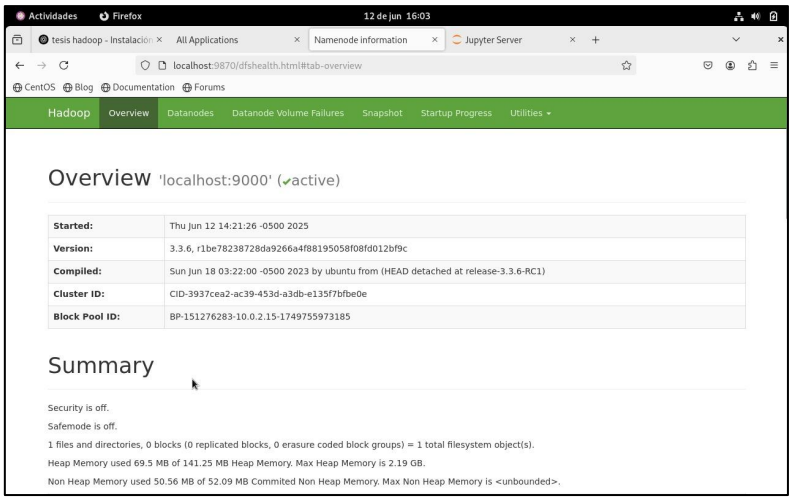
Esta interfaz corresponde al ResourceManager de Hadoop YARN y permite monitorear en tiempo real el estado del clúster.



- **No requiere usuario ni contraseña**: El acceso es directo desde el navegador ingresando http://localhost:8088.
- **Cluster Metrics**: Muestra el estado general del clúster, como aplicaciones enviadas, pendientes, en ejecución, completadas y contenedores activos.
- **Cluster Nodes Metrics**: Indica cuántos nodos están activos, cuántos están siendo retirados y otros estados relacionados a los nodos del clúster.
- **Scheduler Metrics**: Informa sobre el planificador en uso (CapacityScheduler) y los tipos de recursos considerados (memoria y núcleos virtuales).
- **Tabla de Aplicaciones**: Despliega información de cada aplicación procesada por YARN, con columnas

Yarn

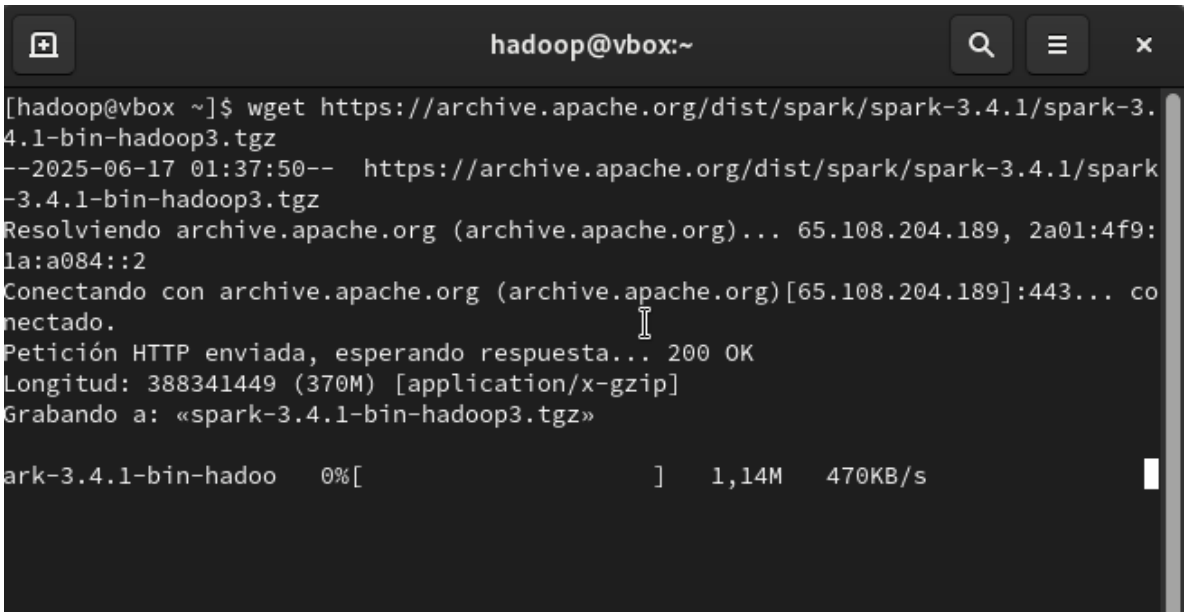
Esta interfaz corresponde al **NameNode de Hadoop HDFS**, accesible desde el navegador mediante http://localhost:9870. Su función principal es mostrar el estado general del sistema de archivos distribuido de Hadoop (HDFS), permitiendo a los usuarios visualizar detalles como cuándo se inició el servicio, la versión de Hadoop instalada, identificadores del clúster y métricas del sistema



- No requiere usuario ni contraseña para acceder.
- Muestra si el NameNode está activo y en funcionamiento.
- Presenta la versión de Hadoop HDFS instalada.
- Indica el Cluster ID y el Block Pool ID del sistema.
- Resume el número total de archivos, bloques y directorios en HDFS.
- Informa si el sistema está en modo seguro (safemode) o si la seguridad está activada.
- Muestra el uso de memoria (heap y no heap).

Instalación del spark

La imagen muestra el proceso de descarga de Apache Spark utilizando el comando wget en una terminal Linux. En este caso, se está descargando el archivo comprimido spark-3.4.1-bin-hadoop3.tgz directamente desde los servidores de Apache, lo cual es una práctica común para obtener la distribución oficial de Spark ya preconfigurada para trabajar con Hadoop 3



- wget https://archive.apache.org/dist/spark/spark-3.4.1/spark-3.4.1-bin-hadoop3.tgz
- tar -xvzf spark-3.4.1-bin-hadoop3.tgz
- sudo mv spark-3.4.1-bin-hadoop3 /opt/spark.

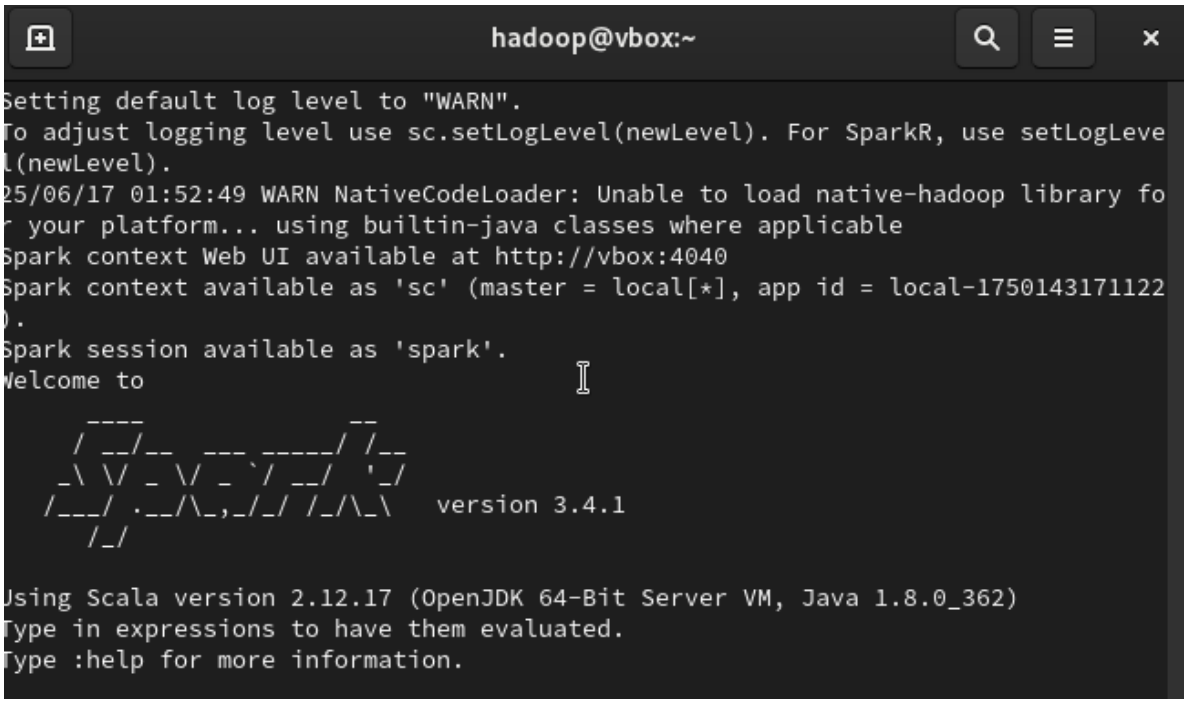
configuración del bash

La configuración de variables de entorno dentro del archivo .bashrc, necesarias para que el sistema operativo reconozca correctamente las rutas de Java, Hadoop y Spark. Esta configuración es crucial porque permite ejecutar comandos como hadoop, spark-shell o pyspark desde cualquier terminal sin tener que especificar rutas completas.

- export SPARK_HOME=/opt/spark
- export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin

spark

Cuando se ejecuta el comando spark-shell, se inicia correctamente Apache Spark en modo consola interactiva. Esto permite ejecutar código en Scala directamente sobre el entorno de Spark. En este caso, Spark detecta la versión 3.4.1 y se ejecuta en modo local (local[*]), utilizando todos los núcleos disponibles del sistema.



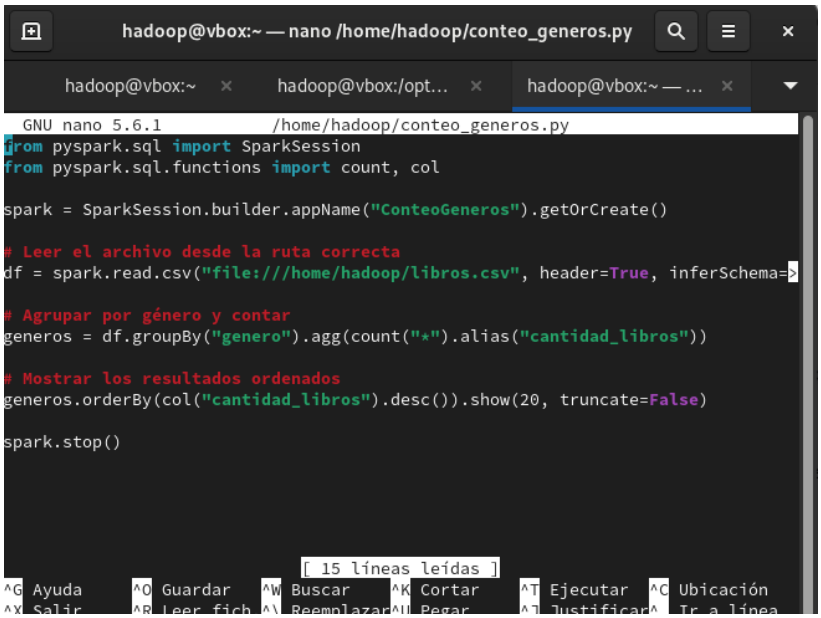
- spark-shell

6. Validación y Pruebas

Se realizaron pruebas utilizando una base de datos pequeña en formato CSV, enfocada en la aplicación de un modelo de árbol de decisiones. El objetivo de las pruebas fue clasificar y predecir géneros de libros en función de los autores y otras características del conjunto de datos. Estas pruebas permitieron verificar el correcto funcionamiento del entorno distribuido, evaluando el rendimiento durante el entrenamiento del modelo, así como la respuesta del sistema en cuanto al uso de CPU y su capacidad de escalar con tareas básicas de machine learning.

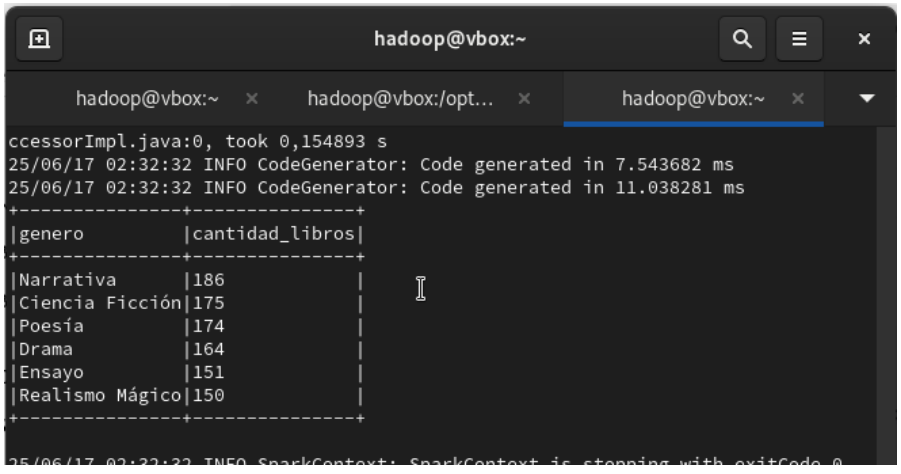
Prueba uno

El script en PySpark permite procesar un archivo CSV llamado libros.csv para contar cuántos libros hay por cada género. Utiliza una sesión de Spark para leer los datos, agruparlos por la columna "género" y aplicar una función de conteo, generando una nueva columna llamada "cantidad libros". Luego, ordena los resultados en orden descendente y los muestra por pantalla



Resultado

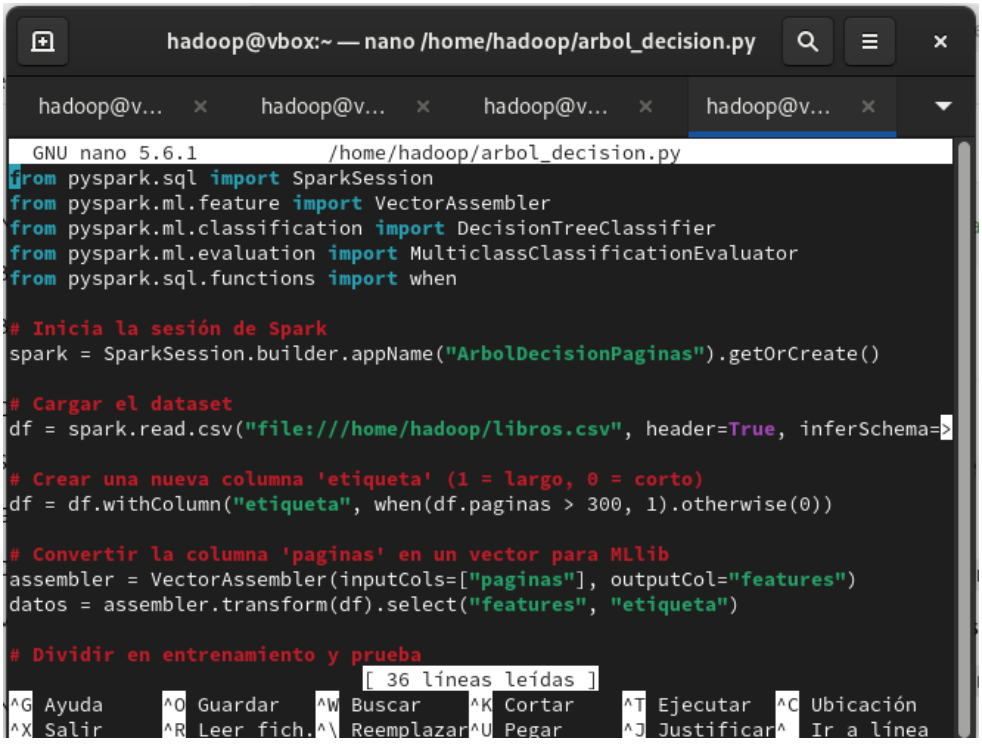
El resultado mostrado corresponde a la salida del análisis realizado con PySpark, donde se agruparon los libros por género y se contó cuántos pertenecen a cada categoría. Se observa que el género con más libros es "Narrativa" con 186 registros, seguido de "Ciencia Ficción" con 175 y "Poesía" con 174. Otros géneros como "Drama", "Ensayo" y "Realismo Mágico" también tienen una representación significativa. Esta salida demuestra que el procesamiento y análisis del dataset se realizó correctamente, y permite identificar cuáles son los géneros más frecuentes dentro del conjunto de datos evaluado.

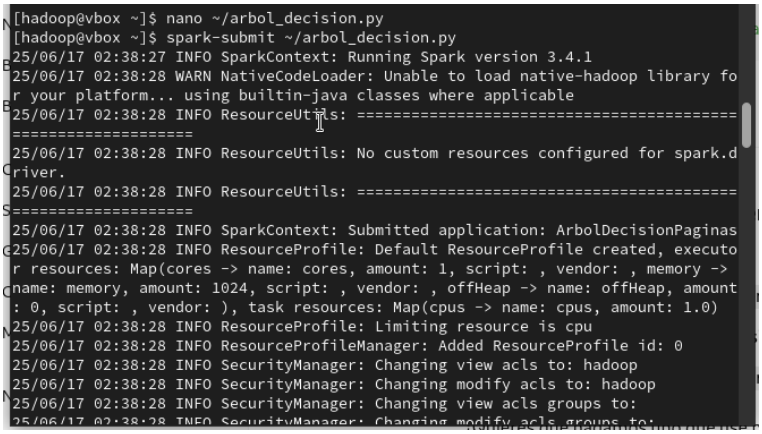


Segunda prueba

Las imágenes corresponden al entrenamiento de un modelo de árbol de decisión usando PySpark. El primer archivo (arbol_decision.py) contiene el script que carga un dataset de libros, crea una columna llamada “etiqueta” donde clasifica los libros como largos o cortos según su número de páginas (mayor a 300 o no), y luego

convierte esta información en vectores de características para ser procesados por el modelo. Posteriormente, el conjunto de datos se divide en entrenamiento y prueba. En la segunda imagen se observa la ejecución de este script mediante el comando spark-submit, lo que inicia el proceso de entrenamiento del modelo sobre el entorno distribuido de Spark, aprovechando los recursos del sistema y validando la correcta integración de las herramientas instaladas

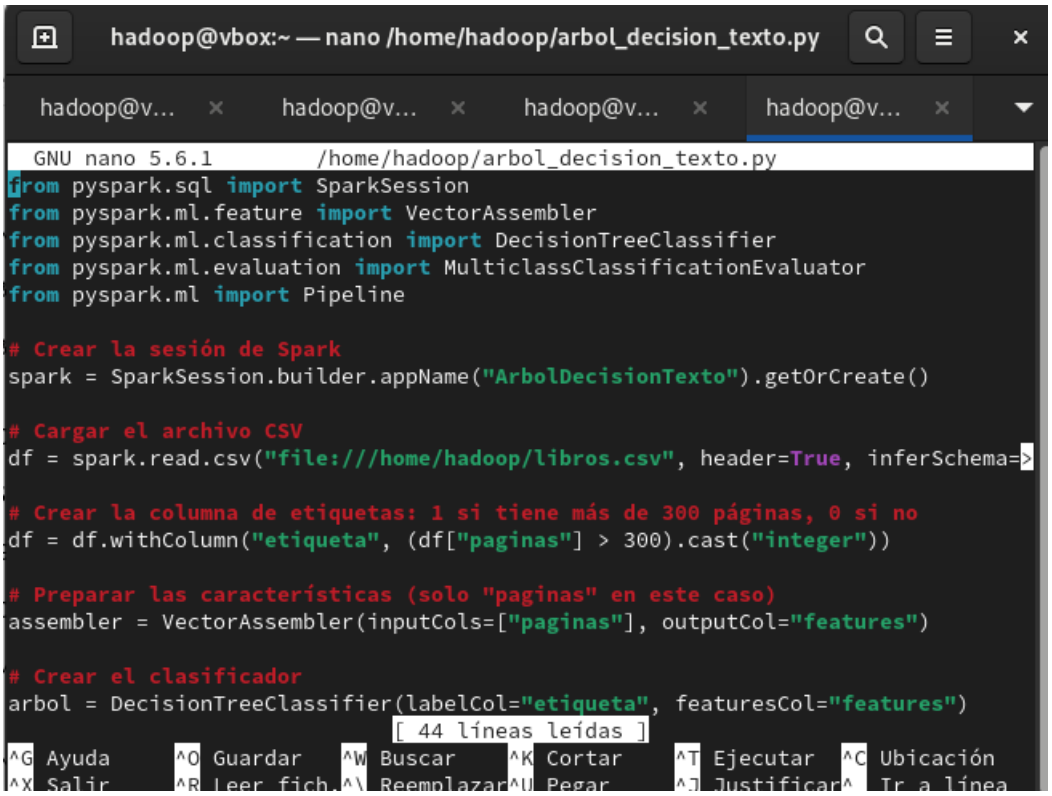




Resultados del entrenamiento

Las dos imágenes corresponden a la ejecución y resultado final del entrenamiento de un modelo de árbol de decisión usando PySpark. En el script, se carga un archivo CSV con datos de libros, se crea una columna "etiqueta" que clasifica los libros como largos (1) si tienen más de 300 páginas o cortos (0) si no, y se utiliza esa información para entrenar un modelo. El árbol de decisión resultante tiene una estructura sencilla:

evalúa si el número de páginas es menor o igual a 307.5 para predecir 0 (libro corto) o 1 (libro largo). Este resultado confirma que el modelo fue entrenado correctamente y es capaz de clasificar nuevas instancias basándose en la cantidad de páginas, demostrando el correcto funcionamiento del entorno de entrenamiento distribuido.




```
Árbol de decisión entrenado:
DecisionTreeClassificationModel: uid=DecisionTreeClassifier_dc07decdd78c, depth=
1, numNodes=3, numClasses=2, numFeatures=1
  If (feature 0 <= 307.5)
   Predict: 0.0
  Else (feature 0 > 307.5)
   Predict: 1.0
```

7. Observaciones Finales

- El sistema es replicable y funcional en laptops personales.
- La arquitectura permite escalar a más nodos si se cuenta con más recursos.
- Se recomienda usar datasets livianos por limitaciones de hardware.