DANE | SEN 2.0 Sistema Estadístico Nacional-Colombia

Guía para la gestión y el mantenimiento del ciclo de vida de modelos de proyectos de ciencia de datos

Departamento Administrativo Nacional de Estadística (DANE)

B. Piedad Urdinola Contreras
Directora

Andrea Ramírez Pisco
Subdirectora

Álvaro Fernando Guzmán Lucero
Secretaria General

Directores técnicos:

Javier Sebastián Ruiz Santacruz
Dirección de Censos y Demografía

Diana María Bohórquez Losada
Dirección de Difusión y Cultura Estadística

Elkin Ernesto Ramírez Niño
Dirección de Geoestadística

César Mauricio López Alfonso
Dirección de Metodología y Producción Estadística

Julieth Alejandra Solano Villa
Dirección de Regulación, Planeación, Estandarización y Normalización

Juan Pablo Cardoso Torres
Dirección de Síntesis y Cuentas Nacionales

Liliana Ibeth Ávila Robles
Encargada de la Dirección de Recolección y Acopio

Luis Martín Barrera Pino
Jefe de Oficina
Oficina de Sistemas

Nelsón Mauricio Parada Botia
Coordinador
GIT Gestión de Datos
Oficina de Sistemas

Alejandro Sandoval Pineda
Líder de Analítica y Ciencia de Datos

Miembros CASEN

Mario Linares Vásquez
Miembro CASEN
Sala de Modernización Tecnológica

Nicolás Cardozo Álvarez
Miembro CASEN
Sala de Modernización Tecnológica

León Darío Parra
Miembro CASEN
Sala de Modernización Tecnológica

Elaboración del documento

Alejandro Sandoval Pineda
Líder de analítica y ciencia de datos
GIT Gestión de datos
Oficina de Sistemas

Consultor conceptual

Daniel Mauricio Montenegro Reyes
Analista de datos
GIT Gestión de datos
Oficina de Sistemas

Revisión

Mario Linares Vásquez
Miembro CASEN

Nicolás Cardozo Álvarez
Miembro CASEN

León Darío Parra
Miembro CASEN

Diseño y diagramación
Jazmin Carolina Waltero Arguello

Corrección de estilo
Sonia Naranjo Morales

CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

SIGLAS Y ACRÓNIMOS

ACID	Atomicity, Consistency, Isolation, and Durability
API	Application Programming Interfaces
ASR	Automatic Speech Recognition
AutoML	Automated Machine Learning
AWS	Amazon Web Services
BI	Business Intelligence
BSD	Berkeley Software Distribution
CASEN	Consejo Asesor Técnico del Sistema Estadístico Nacional
CI/CD	Continuous Delivery and/or Continuous Deployment
CPU	Central Processing Unit
CRISP-DM	Cross-Industry Standard Process for Data Mining
DAG	Directed Acyclic Graph
DANE	Departamento Administrativo Nacional de Estadística
DevOps	Development Operations
DL	Deep Learning
DVC	Data Version Control
EMR	Elastic Map Reduce
ETL	Extract, Transform and Load
FT	Fine Tuning
GPU	Graphical Processing Unit
HDFS	Hadoop Distributed File System

IA	Inteligencia Artificial
IaaS	Infrastructure as a Service
IDE	Integrated Development Environment
KDD	Knowledge Discovery in Databases
LLM	Large Language Model
MLOps	Machine Learning Operations
NLP	Natural Language Processing
OCR	Optical Character Recognition
PaaS	Platform as a Service
RAM	Random Access Memory
RDBMS	Relational Database Management System
RL	Reinforcement Learning
SaaS	Software as a Service
SEMMA	Sample, Explore, Modify, Model, Assess
SEN	Sistema Estadístico Nacional
SQL	Structured Query Language
TDSP	Team Data Science Process
TI	Tecnologías de la Información
TL	Transfer Learning
TPU	Tensor Processing Unit
VCS	/ersion Control System

INTRODUCCIÓN

En octubre de 2023, entró en vigor la Ley 2335 del 2023 en Colombia, la cual establece disposiciones sobre la gestión y la producción de estadísticas oficiales en el país. Particularmente, en el artículo 14 de esta ley se establece la creación del Consejo Asesor Técnico del Sistema Estadístico Nacional (CASEN). Este tiene como objeto asesorar al Departamento Administrativo Nacional de Estadística (DANE) y al Gobierno Nacional sobre cuestiones de importancia estratégica para las estadísticas oficiales del país. A raíz de este hecho, al comienzo de 2024 se establecieron los planes de trabajo para las líneas de investigación que conformarían las salas especializadas relacionadas a cinco temas: salud, bienestar y demografía, gobierno seguridad y justicia, geografía medio ambiente y ordenamiento territorial, economía y modernización tecnológica.

Frente a esta última, la Oficina de Sistemas se determinó como el par técnico del DANE encargado de liderar las salas de modernización tecnológica con dos líneas de investigación correspondientes a ciencia de datos y seguridad y cumplimiento. La línea de ciencia de datos se centró en emplear técnicas y métodos de análisis avanzados para mejorar la recopilación, el procesamiento y la presentación de datos estadísticos producidos desde el DANE. Por su parte, la línea de seguridad y cumplimiento se enfocó en garantizar de manera oportuna la integridad, la privacidad y la confiabilidad de los datos estadísticos generados por el DANE. A su vez, estas líneas de investigación se conformaron por diferentes vertientes correspondientes a: ciclo de vida de modelos de proyectos de ciencia de datos; proyectos de ciencia de datos; anonimización de datos; procesos de interoperabilidad y ciberseguridad, y estándares de seguridad informática con alcance internacional.

El presente documento centraliza en una guía los conceptos, las prácticas y las herramientas predominantes, más avanzadas y de vanguardia en el sector estadístico y empresarial, orientadas a los miembros del SEN, para la gestión y el mantenimiento del ciclo de vida de los modelos en proyectos de ciencia de datos. Lo anterior en respuesta a las necesidades actuales, los desafíos y la visión del DANE relacionada con la modernización, la innovación y la gestión tecnológica. La construcción de esta guía es el resultado de un exhaustivo análisis de múltiples fuentes y referentes relacionados en el campo de ciencia de datos e ingeniería de software. Por un lado, se incluye el enfoque ágil para el desarrollo de proyectos completos de ciencia de datos, siguiendo las metodologías promovidas por la Agile Alliance (Agile Alliance, 2001) para iterar rápidamente, adaptarse a los cambios y entregar valor de manera continua a lo largo del ciclo de vida del proyecto. Por otro lado, se integran las mejores prácticas de ingeniería de software para la operacionalización de modelos de aprendizaje de máquinas (MLOps) para abarcar aspectos como la gestión del ciclo de vida del modelo, la integración y despliegue continuo y la monitorización de modelos en producción (Gift & Deza, 2021).

Esta guía se presenta como un marco referencial integral para el desarrollo de proyectos de ciencia de datos, abarcando todos los componentes esenciales necesarios para asegurar su éxito. La guía incluye una descripción detallada de la arquitectura de datos, la infraestructura y las herramientas y los artefactos requeridos en cada etapa del proyecto discriminados por el nivel de madurez y complejidad del proyecto de ciencia de datos. Adicionalmente se detallan casos de prueba, junto con ejemplos ilustrativos de la aplicación de esta guía.

Este documento se compone por cuatro capítulos. En el primero se presentan los aspectos teóricos y conceptuales claves en ciencia de datos, esenciales para la ejecución de proyectos. En el segundo se exponen las infraestructuras y las herramientas más utilizadas, tanto en el ámbito académico como en la producción para la implementación de proyectos de ciencia de datos. En el tercero se expone la guía para la gestión y el mantenimiento del ciclo de vida de modelos de proyectos de ciencia de datos. En el cuarto se expone un caso de prueba ilustrativo de la aplicación de la guía.

OBJETIVOS Y **ALCANCE** DE LA GUÍA

Los objetivos específicos del documento son:

1. Proporcionar un glosario exhaustivo de conceptos clave en ciencia de datos que son esenciales para la ejecución de proyectos.

2. Centralizar las infraestructuras y las herramientas más avanzadas y prevalentes, tanto en el ámbito académico como en la producción, y que son utilizadas para la implementación de proyectos de ciencia de datos.

3. Presentar las mejores prácticas de ingeniería de software para asegurar la correcta implementación de las diversas etapas del ciclo de vida de un modelo de ciencia de datos.

4. Detallar los artefactos asociados a cada etapa de un proyecto de ciencia de datos y definir estándares claros para los contenidos de cada uno de ellos.

5. Ejemplificar la implementación de esta guía mediante un caso de prueba práctico.

El alcance de esta guía se delimita a brindar una herramienta clara y estructurada que abarque los conceptos, metodologías y herramientas más avanzadas y predominantes en el ámbito estadístico y empresarial, enfocadas en la gestión del ciclo de vida de modelos en proyectos de ciencia de datos. El contenido de esta obedece al consenso entre la Oficina de Sistemas, que actuó como par técnico del DANE y líder de las salas de modernización tecnológica, y los miembros expertos del CASEN. Lo anterior con el fin de garantizar que la guía no solo refleja el rigor técnico necesario, sino también las mejores prácticas actualmente acordadas en el sector, asegurando su aplicabilidad efectiva por parte de los actores y entes del Sistema Estadístico Nacional (SEN).

1.
ASPECTOS **TEÓRICOS Y CONCEPTUALES**

Este capítulo aborda los aspectos teóricos y conceptuales relacionados con el ciclo de vida de modelos en proyectos de ciencia de datos. Se presentan definiciones técnicas clave que forman el marco teórico necesario para que el lector de esta guía cuente con el conocimiento de base fundamental en el campo de la ciencia de datos.

1.1.Glosario de conceptos relacionados con ciencia de datos

Ciencia de datos

Es un campo interdisciplinario que utiliza técnicas, métodos y herramientas para analizar, interpretar y extraer conocimiento útil de conjuntos de datos, con el objetivo de tomar decisiones informadas y resolver problemas complejos en una amplia variedad de dominios (Hastie et al., 2009)

A continuación, se presenta el glosario de conceptos básicos que están directamente relacionados con ciencia de datos. Esto como parte fundamental para establecer una base sólida estandarizada de conceptos que se puedan interiorizar por equipos de ciencia de datos en la Entidad. Para efectos de la presente guía se adoptarán las siguientes definiciones[1]:

Ajuste fino *(Fine Tuning – FT):* implica tomar un modelo pre entrenado y ajustar o afinar sus pesos utilizando un nuevo conjunto de datos específicos. En el FT se suelen congelar algunas capas del modelo origen (pre entrenado) para retener conocimientos previos y eliminar las capas originales completamente conectadas para agregar otro clasificador, ya sea mediante una capa completamente conectada u otro clasificador que permita la extracción de características (Tan et al., 2018; Weiss et al., 2016).

Analítica de datos: en el ámbito científico y tecnológico, la analítica se refiere al proceso de examinar conjuntos de datos grandes y complejos con el objetivo de descubrir patrones, tendencias y relaciones que puedan proporcionar información valiosa para la toma de decisiones. Implica el uso de técnicas estadísticas, matemáticas y computacionales para realizar análisis exploratorios, modelado predictivo y otras formas de análisis de datos avanzados (Provost & Fawcett, 2013).

Análisis Exploratorio de Datos *(Exploratory Data Analysis – EDA):* es un enfoque utilizado para analizar conjuntos de datos y resumir sus principales características mediante métodos visuales y estadísticas descriptivas. Su objetivo es descubrir patrones, detectar anomalías, verificar suposiciones y formular hipótesis iniciales (Tukey, 1977).

Aprendizaje de máquina (Machine Learning – ML): es un campo de la inteligencia artificial que se centra en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender a partir de datos y mejorar con la experiencia, sin necesidad de programación explícita para realizar una tarea (Hastie et al., 2009).

Aprendizaje supervisado: enfoque dentro del campo del aprendizaje de máquina donde se entrena un modelo utilizando un conjunto de datos etiquetado, es decir, que tienen pares de entrada y salida esperada. Aborda dos tipos principales de problemas regresión y clasificación (Bishop, 2006).

Aprendizaje no supervisado: enfoque dentro del campo del aprendizaje de máquina donde el modelo se entrena utilizando datos que no están etiquetados, es decir, el modelo descubre patrones o estructuras inherentes en los datos por sí mismo. Algunos de los problemas comunes que aborda son el agrupamiento (clustering), reducción de dimensionalidad y detección de anomalías (Bishop, 2006).

Aprendizaje profundo *(Deep Learning – DL:* subárea del aprendizaje de máquina que se centra en el entrenamiento de modelos de redes neuronales profundas para aprender representaciones complejas de datos. Utiliza algoritmos de aprendizaje supervisado o no supervisado para aprender características complejas y abstracciones de datos de entrada (Schmidhuber, 2015).

Aprendizaje por refuerzo *(Reinforcement Learning - RL):* enfoque en el campo del aprendizaje de máquina donde un agente aprende a tomar acciones en un entorno con el objetivo de maximizar una recompensa acumulativa a lo largo del tiempo. En este paradigma, el agente interactúa con un entorno, observa el estado del entorno, toma una acción y recibe una recompensa (positiva o negativa) en función de la acción tomada y el estado resultante del entorno (Sutton & Barto, 2018).

Aprendizaje de Transferencia *(Transfer Learning - TL):* es un área importante en el campo del aprendizaje de máquina que consiste en transferir conocimiento de un modelo pre entrenado en una tarea específica y adaptar o transferir su conocimiento

[1] Se aclara que las definiciones sin referencia explícita se consideran términos estándar que gozan de amplio reconocimiento y aceptación dentro de la comunidad profesional en el ámbito de la ciencia de datos y la industria.

a una tarea diferente pero relacionada (Tan et al., 2018; Weiss et al., 2016).

Automatización: uso de tecnologías para realizar procesos o tareas con mínima intervención humana, implementando sistemas basados en reglas predefinidas, algoritmos o programas de software que buscan aumentar la eficiencia, productividad y fiabilidad.

Backlog: lista estructurada de tareas o requisitos que un equipo de desarrollo debe completar. Cada elemento del backlog, denominado backlog item o work item, se organiza y prioriza para ser asignado a iteraciones específicas, como sprints. Estos elementos se agrupan según su nivel de granularidad en épics, features o historias de usuario, asegurando un enfoque incremental y ajustable que maximiza la eficiencia y el valor entregado al cliente durante el ciclo de desarrollo (Atlassian, 2024f).

Canalización: secuencia de pasos o procesos interconectados diseñados para procesar datos o ejecutar tareas de manera eficiente y automatizada, donde cada paso realiza una función específica y los resultados de un paso se utilizan como entrada para el siguiente, permitiendo que los datos fluyan de manera continua.

Ciclo de vida de un modelo de ciencia de datos: conjunto de etapas y procesos involucrados en el desarrollo, la implementación, la evaluación y el mantenimiento de modelos de ciencia de datos. Estas etapas típicamente incluyen la definición del problema, la recopilación y la preparación de datos, la selección y el entrenamiento del modelo, la evaluación del rendimiento del modelo y la implementación en un entorno de producción (Provost & Fawcett, 2013).

Conocimientos: comprensión, conciencia o familiaridad obtenida a partir de la experiencia, educación o aprendizaje. Se construyen sobre la información al combinarla con experiencias previas, interpretación y habilidades analíticas, permitiendo la aplicación práctica y la toma de decisiones informadas.

Datos: representaciones simbólicas de hechos y figuras observables, recopilados y registrados para su análisis. No tienen significado intrínseco hasta que se interpretan dentro de un contexto específico.

Despliegue continuo (CD): extensión de la integración continua que automatiza la entrega del código a un entorno de producción. Cada cambio que pasa las pruebas automatizadas es automáticamente desplegado en producción, permitiendo que nuevas funcionalidades y correcciones lleguen a los usuarios de manera rápida y confiable.

Development Operations (DevOps): metodología de desarrollo de software que promueve la colaboración entre los equipos de desarrollo (Dev) y operaciones (Ops), con el objetivo de automatizar el proceso de entrega de software y mejorar la calidad y velocidad de implementación de aplicaciones (Bass et al., 2015).

Información: resultado del procesamiento, la organización y la estructuración de datos para darles contexto y significado. Proporciona respuestas a preguntas como quién, qué, cuándo, dónde y cómo, transformando los datos en algo útil para la toma de decisiones.

Infraestructura como código *(Infraestructure as Code – IaC):* práctica de gestión y aprovisionamiento de la infraestructura de TI mediante archivos de configuración que describen los recursos y configuraciones necesarios, en lugar de usar configuraciones manuales o herramientas de gestión de infraestructura tradicionales. Estos archivos pueden ser versiones controladas, revisados y reutilizados, permitiendo la automatización, consistencia y replicabilidad de entornos de desarrollo, pruebas y producción.

Integración continua (CI): es una práctica de desarrollo de software en la que desarrolladores integran su código en un repositorio central de manera frecuente, al menos una vez al día. Cada integración es verificada por una compilación automatizada y pruebas, lo que permite detectar errores

rápidamente y mejorar la calidad del software.

Inteligencia artificial *(Artificial Intelligence - IA):* campo de las ciencias de la computación que se centra en la creación de sistemas inteligentes que pueden simular el comportamiento humano, aprender de la experiencia y realizar tareas complejas de manera autónoma (Russell & Norvig, 2016).

Inteligencia artificial general *(Artificial General Intelligence – AGI):* forma de inteligencia artificial que posee la capacidad de comprender, aprender y aplicar conocimientos en una amplia gama de tareas al nivel de un ser humano. La AGI puede transferir su aprendizaje y habilidades de una tarea a otra, demostrando un alto grado de adaptabilidad y resolución de problemas (Legg & Hutter, 2007).

Inteligencia artificial estrecha *(Narrow AI):* también conocida como IA débil, se refiere a sistemas de inteligencia artificial diseñados y entrenados para realizar una tarea específica o un conjunto limitado de tareas (Goertzel, 2007).

Inteligencia artificial generativa: enfoque dentro del campo de la inteligencia artificial que se centra en el desarrollo de sistemas capaces de crear contenidos de manera autónoma tales como imágenes, música, texto o incluso videos, imitando el estilo y la estructura de los datos de entrada (Goodfellow et al., 2016).

Inteligencia artificial responsable *(IA responsable):* enfoque para desarrollar, evaluar e implementar sistemas de IA de manera segura, confiable y ética acorde con seis principios: equidad, confiabilidad y seguridad, privacidad y seguridad, inclusión, transparencia y responsabilidad (OCDE, 2019).

Machine Learning Operations – MLOps: metodología que combina prácticas de desarrollo de software (DevOps) con procesos específicos para el desarrollo, implementación y operación de sistemas de aprendizaje automático (ML). Tiene como objetivo mejorar la colaboración y la eficiencia entre los equipos de ciencia de datos, desarrollo y operaciones, mediante la automatización de flujos de trabajo, la gestión del ciclo de vida de los modelos, el monitoreo del rendimiento y la escalabilidad de los modelos de ML en entornos de producción (Machine Learning for Developers, 2022).

Modelo: representación abstracta, conceptual, gráfica o matemática de sistemas, fenómenos o procesos del mundo real. Los modelos se utilizan en una amplia variedad de disciplinas, como la ciencia, la ingeniería, la economía y la informática, para comprender, simular y predecir comportamientos y resultados (Stachowiak, 1973).

Modelo de ciencia de datos: representación simplificada o abstracción de un fenómeno del mundo real mediante una función matemática o estadística que mapea las variables de entrada a una variable de salida. Este puede ser de tipo exploratorio, descriptivo, diagnostico, causal, prescriptivo o predictivo permitiendo responder preguntas como: ¿Qué puede descubrirse?, ¿Qué pasó?, ¿Qué está sucediendo?, ¿Por qué pasó?, ¿Qué debería hacer? y ¿Qué pasará? del fenómeno en estudio (Hastie et al., 2009).

Modelo estadístico: representación matemática o probabilística de un fenómeno del mundo real que se utiliza para describir, analizar o predecir datos (Moore et al., 2014).

Modelo de aprendizaje de máquinas: representación matemática o computacional que se entrena utilizando datos para realizar tareas específicas, como clasificación, regresión, o agrupamientos, entre otras, sin ser explícitamente programado para realizar esas tareas (Hastie et al., 2009).

Modelo de aprendizaje profundo: tipo de modelo de aprendizaje de máquinas que utiliza redes neuronales artificiales con múltiples capas ocultas para aprender representaciones jerárquicas de datos (Goodfellow et al., 2016).

Modelo de procesamiento de lenguaje natural: tipo de modelo computacional di-

señado para comprender lenguaje humano de manera efectiva utilizando técnicas de aprendizaje de máquinas y procesamiento del lenguaje natural permitiendo analizar, interpretar y generar texto de manera automática (Bird et al., 2009).

Modelos de lenguaje grande *(Large Language Model – LLM):* es un tipo de modelo de procesamiento del lenguaje natural que utiliza una gran cantidad de parámetros y datos para aprender representaciones de texto a gran escala. Está diseñado para comprender y generar texto de manera efectiva, capturando las complejidades y sutilezas del lenguaje humano (Vaswani et al., 2017).

Modelo exploratorio: tipo de modelo utilizado para investigar datos sin hipótesis preconcebidas, con el objetivo de descubrir patrones, relaciones y tendencias. Su propósito es generar nuevas hipótesis y obtener una comprensión inicial de los datos, permitiendo identificar posibles áreas de interés para estudios más detallados.

Modelo descriptivo: tipo de modelo que se enfoca en resumir y describir las principales características de un conjunto de datos mediante estadísticas básicas y gráficos. Su objetivo es proveer una visión general y comprensible de los datos, facilitando el entendimiento de las distribuciones y comportamientos generales.

Modelo relacional: tipo de modelo que examina las relaciones entre variables o factores para entender cómo están conectadas e influyen entre sí. Su propósito es identificar y cuantificar las interdependencias entre las variables, proporcionando una base sólida para la comprensión de las estructuras subyacentes en los datos.

Modelo causal: tipo de modelo que busca identificar y establecer relaciones de causa y efecto entre variables mediante métodos estadísticos y experimentales. Su propósito es determinar por qué ocurren ciertos fenómenos y cómo los cambios en una variable afectan a otra, permitiendo la inferencia de causalidad a partir de datos observacionales o experimentales.

Modelo predictivo: tipo de modelo que utiliza algoritmos para predecir resultados futuros basados en datos históricos. Su objetivo es proveer predicciones precisas para aplicaciones como previsión de ventas, detección de fraudes y mantenimiento predictivo, ayudando a tomar decisiones informadas basadas en proyecciones.

Modelo prescriptivo: tipo de modelo que genera recomendaciones sobre acciones óptimas y sus posibles impactos utilizando simulaciones y algoritmos de optimización. Su propósito es apoyar la toma de decisiones estratégicas y operativas, optimizando los resultados según los objetivos establecidos, y proporcionando guías claras para la implementación de estrategias efectivas.

Procesamiento de Lenguaje Natural *(Natural Language Processing - NLP):* es un campo interdisciplinario de la inteligencia artificial y la lingüística computacional que tiene como objetivo permitir a los computadores comprender e interpretar el lenguaje natural en forma legible. Abarca una variedad de tareas que incluyen el análisis de sentimientos, traducción automática, extracción de información, generación de lenguaje natural, entre otros (Bird et al., 2009).

Reconocimiento óptico de caracteres *(Optical Character Recognition - OCR):* es un campo de la inteligencia artificial encargado del procesamiento de imágenes y archivos utilizado para identificar y extraer caracteres individuales o cadenas de caracteres, permitiendo su conversión en formato digital y su posterior manipulación por computador (Tesseract OCR Team, 2020).

Reconocimiento automático del habla *(Automatic Speech Recognition - ASR):* es un campo interdisciplinario de la inteligencia artificial, procesamiento de señales y lingüística computacional que permite convertir señales de audio en texto escrito de manera automática (Huang et al., 2001).

Red Neuronal Artificial *(Artificial Neural Network – ANN):* es un modelo computacional inspirado en la estructura y funcionamiento de las redes neuronales biológicas. Consiste en un conjunto de nodos (o neu-

ronas) interconectados, donde cada nodo realiza una operación de procesamiento simple. Las conexiones entre nodos tienen pesos que se ajustan durante el proceso de aprendizaje para optimizar el rendimiento del modelo (Schmidhuber, 2015).

Replicabilidad: capacidad de obtener los mismos resultados que un estudio original utilizando la misma metodología, datos y condiciones, pero realizado por investigadores diferentes.

Reproducibilidad: capacidad de obtener resultados consistentes al repetir un análisis de datos con los mismos métodos y condiciones, pero potencialmente con nuevos datos o en un entorno diferente.

Súperinteligencia: forma de inteligencia artificial que no solo iguala, sino que supera la inteligencia humana en todos los aspectos cognitivos, incluidos la creatividad, la resolución de problemas y la toma de decisiones. Una superinteligencia es capaz de realizar tareas que son imposibles para los seres humanos y de hacerlo con una eficiencia y velocidad mucho mayores (Bostrom, 2014).

Visión de computador: campo de la inteligencia artificial que permiten obtener información a partir de imágenes y videos digitales. Esto incluye técnicas para el procesamiento y análisis de imágenes, detección de objetos, segmentación semántica, así como la comprensión de la escena visual en general (Szeliski, 2010).

1.2. Conceptos de infraestructura para proyectos de ciencia de datos

Infraestructura para ciencia de datos

La infraestructura comprende el conjunto de hardware, software y redes necesarios para soportar el procesamiento, almacenamiento y análisis de datos.

A continuación, se detallan los conceptos clave de infraestructura de hardware y software esenciales para el desarrollo de proyectos de ciencia de datos, abarcando los componentes críticos que permiten una operación eficiente y eficaz en cada etapa del ciclo de vida del proyecto:

Base de datos: conjunto de datos organizados y almacenados de forma sistemática en un sistema informático permitiendo la gestión y la recuperación de datos, así como la generación de consultas y el análisis de la información almacenada.

Base de datos multidimensional *(Muldimensional Data Base – MDB):* tipo de base de datos optimizada para aplicaciones de inteligencia empresarial y procesamiento analítico en línea (OLAP). Las MDB organizan datos en una estructura multidimensional, donde cada dimensión representa un aspecto diferente de los datos, como tiempo, ubicación o producto.

Base de datos relacional: tipo de base que organiza la información en tablas estructuradas con filas (registros) y columnas (atributos), y relaciones entre estas, estas últimas son definidas a partir del uso de claves primarias y foráneas, facilitando la integridad y consistencia de los datos.

Base de datos no relacional (NoSQL): tipo de sistema de gestión de bases de datos diseñado para almacenar, recuperar y gestionar datos no estructurados o semiestructurados, que no se ajustan al modelo de tablas relacionales. Estas bases de datos utilizan diversos modelos de datos como documentos, clave-valor, columnas anchas y grafos, para proporcionar flexibilidad y escalabilidad en el manejo de grandes volúmenes de datos.

Bodega de datos *(data warehouse):* infraestructura de almacenamiento de datos diseñada para almacenar grandes volúmenes de datos de diversas fuentes en un formato optimizado para el análisis y la generación de informes. Su objetivo principal es proporcionar un repositorio centralizado y consolidado de datos que permite realizar consultas complejas y análisis en tiempo real.

Clúster: conjunto de computadores/servidores, también conocidos como nodos, interconectados a través de una red que trabajan juntos como si fueran una única entidad para realizar tareas específicas de manera eficiente y escalable. Son utilizados para distribuir cargas de trabajo y mejorar el rendimiento en el procesamiento de gran des volúmenes de datos mediante la división de tareas en unidades más pequeñas que pueden ejecutarse de forma simultánea.

Computación on-premise: modelo de implementación de tecnología de la información en el que los recursos informáticos, incluidos servidores, almacenamiento y software, se alojan y mantienen localmente en las instalaciones físicas de una organización.

Computación en la nube: modelo de entrega de servicios de tecnología de la información a través de Internet, donde los recursos informáticos, como servidores, almacenamiento, bases de datos, redes, software y otros servicios, se ofrecen y se acceden de manera remota a través de la red, generalmente a través de un proveedor de servicios en la nube.

Contenedor: unidad estandarizada de software que empaqueta el código de una aplicación y todas sus dependencias, permitiendo que se ejecute de manera consistente en cualquier entorno. Se caracteriza por su portabilidad, ligereza y aislamiento, son rápidos de iniciar, fáciles de escalar horizontalmente, y son ideales para entornos de Integración Continua y Despliegue Continuo (CI/CD).

CPU (Unidad de Procesamiento Central *(Central Processing Unit – CPU)*): es la encargada de ejecutar instrucciones de programas mediante operaciones aritméticas, lógicas, de control y de entrada/salida. Aunque versátil y fundamental para tareas generales, su capacidad de procesamiento paralelo es limitada, lo que reduce su eficiencia en el entrenamiento de modelos de ciencia de datos intensivos en cálculos.

Cubo de Procesamiento Analítico en Línea *(On-Line Analytical Processing – OLAP):* es una estructura de datos multidimensional que permite realizar análisis complejos y consultas rápidas sobre grandes volúmenes de datos. Facilita la visualización de datos desde múltiples perspectivas y dimensiones.

Cubo de Procesamiento Analítico Multidimensional en Línea *(Multidimensional Online Analytical Processing – MOLAP):* arquitectura OLAP que utiliza bases de da-

tos multidimensionales, en la que la información se almacena para ser visualizada en varias dimensiones de análisis.

Cubo de Procesamiento Analítico Relacional en Línea *(Relational Online Analytical Processing – ROLAP):* arquitectura OLAP que permite el análisis multidimensional de datos almacenados en bases de datos relacionales mediante consultas SQL para crear vistas multidimensionales de los datos de forma dinámica.

Dashboard: interfaz gráfica que presenta de manera visual información clave, métricas o indicadores de rendimiento de un sistema, proceso o conjunto de datos específicos. Suelen mostrar datos en forma de gráficos, tablas, indicadores y otros elementos visuales, permitiendo entender rápidamente el estado y la tendencia de los datos y tomar decisiones informadas en función de esa información.

Data Mart: sistema de almacenamiento de datos que contiene información específica de una unidad de negocio, departamento o área de una organización. Contiene una parte pequeña y específica de los datos que la empresa aloja en un sistema de almacenamiento más grande como un Datawarehouse.

Data Mesh: enfoque arquitectónico para la gestión de datos que promueve la descentralización mediante la organización de los datos en dominios específicos, donde cada equipo es responsable de la gestión, calidad, y disponibilidad de sus propios conjuntos de datos.

Hardware: comprende los componentes físicos de un sistema informático que incluyen dispositivos electrónicos, circuitos, cables, periféricos y cualquier otro elemento tangible que constituya la infraestructura física de un sistema informático. Esto abarca desde los componentes internos de una computadora, como la unidad central de procesamiento (CPU), la memoria RAM, el disco duro y las unidades de procesamiento gráfico GPU, hasta los dispositivos periféricos externos, como monitores, teclados, ratones, impresoras y dispositivos de almacenamiento externo.

Infraestructura como un servicio *(IaaS):* modelo de computación en la nube que proporciona el acceso a recursos informáticos escalables, como servidores virtuales, redes, almacenamiento y otros recursos de infraestructura, a través de Internet. En este modelo, los proveedores de servicios en la nube gestionan y mantienen la infraestructura subyacente, mientras que los usuarios tienen control sobre la configuración, el despliegue y la gestión de los recursos virtuales según sus necesidades.

Lago de datos *(data lake):* es un repositorio centralizado con capacidades para la ingesta, el almacenamiento y el procesamiento de datos heterogéneos, tanto estructurados como no estructurados, en su forma nativa, lo que permite su análisis de manera ágil y flexible.

Máquina virtual *(Virtual Machine – VM):* entorno informático que emula una computadora física, funcionando como un sistema aislado con su propia CPU, GPU, memoria, interfaz de red y almacenamiento, creado a partir de recursos de hardware compartidos gestionados por una una capa de software conocida como hipervisor.

Orquestador: herramienta o software que se encarga de coordinar y gestionar el despliegue, ejecución y escalado de aplicaciones y servicios en entornos distribuidos y/o en la nube. Automatiza y administra las tareas necesarias para garantizar que los componentes de una aplicación se ejecuten correctamente, distribuyendo cargas de trabajo entre los diferentes nodos o máquinas del sistema de manera eficiente y escalable. Esto incluye la asignación de recursos, la gestión de la comunicación entre los componentes, el monitoreo del estado de los servicios y la gestión de la escalabilidad según la demanda.

Piscina de datos *(data pool):* repositorio que permite el almacenamiento, la centralización y la organización de datos de diversas fuentes. Es escalable y optimizado para

consultas y análisis de datos predefinidos, ideal para casos de uso donde se requiere un acceso rápido a datos estructurados y preprocesados listos para ser consumidos por aplicaciones.

Plataforma como un servicio *(PaaS):* modelo de computación en la nube que proporciona un entorno de desarrollo y despliegue completo en la nube para construir, ejecutar y gestionar aplicaciones sin la complejidad de administrar la infraestructura subyacente. En este modelo, los proveedores de servicios en la nube ofrecen una plataforma que incluye sistemas operativos, entornos de ejecución, bases de datos, herramientas de desarrollo, integración, seguridad y otros servicios, accesibles a través de internet. Los usuarios pueden desarrollar, probar, implementar y escalar aplicaciones de forma rápida y eficiente, centrándose en el desarrollo de aplicaciones y la innovación, mientras que el proveedor de servicios en la nube se encarga de la gestión y mantenimiento de la infraestructura y la plataforma.

Software: abarca programas informáticos, scripts y datos asociados que proporcionan instrucciones y funciones para que los dispositivos informáticos ejecuten tareas específicas. Esto incluye sistemas operativos, aplicaciones de software, herramientas de desarrollo, utilidades y cualquier otro componente digital que pueda ser ejecutado por un sistema informático. El software puede ser clasificado en distintas categorías, como software de sistema, que proporciona servicios básicos para el funcionamiento del hardware y otros programas; y software de aplicación, que se utiliza para realizar tareas específicas según las necesidades del usuario.

Software como un servicio *(SaaS):* modelo de distribución de software en el que las aplicaciones son alojadas por un proveedor de servicios en la nube y están disponibles para los usuarios a través de internet. En este modelo, los usuarios pueden acceder a las aplicaciones a través de un navegador web o una Interfaz de Programación de Aplicaciones (API) sin necesidad de instalar o mantener el software en sus propios dispositivos. El proveedor de servicios en la nube se encarga de la gestión de la infraestructura, la seguridad, las actualizaciones y el mantenimiento del software, mientras que los usuarios pagan por el uso del servicio según un modelo de suscripción, generalmente basado en el número de usuarios o el consumo de recursos.

Unidad de Procesamiento Gráfico *(Graphics Processing Unit – GPU):* unidad de sistema de cómputo diseñada para operaciones altamente paralelas, inicialmente orientada al renderizado de gráficos. Su capacidad para manejar miles de núcleos simultáneamente la convierte en una herramienta poderosa para acelerar el entrenamiento de modelos de aprendizaje profundo, haciendo que sean significativamente más rápidos en comparación con las CPU.

Unidad de Procesamiento Tensorial *(Tensor Processing Unit – TPU):* unidad diseñada para optimizar operaciones de aprendizaje automático, en particular para redes neuronales profundas. Su diseño enfocado en operaciones matriciales y aritmética de precisión reducida la hace superior a las CPU y GPU en tareas de entrenamiento e inferencia de modelos, permitiendo una escalabilidad y velocidad sin precedentes en aplicaciones de inteligencia artificial.

Workers: procesos o hilos dedicados que realizan tareas específicas y paralelas dentro de una aplicación o sistema. Se utilizan para gestionar cargas de trabajo concurrentes, mejorar la eficiencia y la capacidad de respuesta del sistema. Pueden procesar trabajos de una cola de tareas, realizar cálculos en segundo plano, o manejar solicitudes de usuario de manera asíncrona.

1.3. Talento humano, roles y responsabilidades para proyectos de ciencia de datos

Talento humano para proyectos de ciencia de datos

El talento humano es el personal esencial para la ejecución y el mantenimiento eficiente de proyectos de ciencia de datos.

Para el desarrollo y la implementación eficiente de un proyecto de ciencia de datos, es fundamental una definición clara y precisa de los roles y las responsabilidades de cada miembro del equipo. La distribución de responsabilidades depende en gran medida de las habilidades del equipo, y es probable que roles y responsabilidades evolucionen con el tiempo, especialmente a medida que el equipo adopte mejores prácticas en el ciclo de vida de un modelo de ciencia de datos. No obstante, se presenta un conjunto de roles y responsabilidades organizados en cuatro grupos funcionales para el desarrollo de un proyecto de ciencia de datos.

Roles de liderazgo comercial:

- **Gerente de grupo *(Group Manager):*** administra toda la unidad de ciencia de datos en una empresa. Una unidad de ciencia de datos puede tener varios equipos que trabajan en múltiples proyectos de ciencia de datos. Es el responsable de: crear una cuenta asociada al grupo en una plataforma de versionado de código y de establecer la estructura plantilla para el repositorio de documentación del proyecto.

- **Líder de equipo *(Team Lead):*** gestiona un equipo en la unidad de ciencia de datos de una empresa. Para una unidad pequeña de ciencia de datos, el gerente del grupo y el líder del equipo puede ser la misma persona que desempeña las responsabilidades de ambos roles. Es el responsable de configurar el equipo de trabajo a nivel de roles de acuerdo con las habilidades de sus integrantes.

- **Líder de proyecto *(Project Lead o Project Manager):*** gestiona las actividades diarias del equipo en un proyecto de ciencia de datos. Aprovisiona al equipo de trabajo con las plataformas de desarrollo y reposito-

rios de código. Para equipos pequeños de ciencia de datos, este rol suple las responsabilidades del gerente de grupo y líder de equipo. Es el responsable de la planificación y ejecución del proyecto. Aprovisiona al equipo el acceso a las plataformas de desarrollo y versionado de código. Y brinda concepto técnico en decisiones clave sobre el uso de tecnologías, herramientas, marcos de trabajo y bibliotecas de ciencia de datos.

- **Propietario del producto *(Product Owner):*** es el responsable de maximizar el valor del producto desarrollado por el equipo. Actúa como el enlace entre el equipo de desarrollo y las partes interesadas, tomando decisiones sobre el alcance, priorización y aceptación de las entregas. Se encarga de gestionar y priorizar el backlog del producto, asegurando que las características y funcionalidades entregadas se alineen con los objetivos del negocio y las necesidades del cliente. Para equipos pequeños de ciencia de datos, este rol suple las responsabilidades del gerente de grupo, líder de equipo y de proyecto.

- **Analista de negocios *(Business Analyst):*** es el encargado de identificar, documentar y analizar los requisitos del negocio y asegurar que estos se traduzcan en soluciones efectivas y alineadas con los objetivos del proyecto. Aunque su rol puede solaparse con el de un *Product Owner* en algunos aspectos, el analista de negocio se enfoca más en el análisis detallado de procesos y requisitos, colaborando con el dueño del producto para garantizar que las necesidades del cliente se entiendan y se implementen correctamente.

Roles de ciencia de datos, aprendizaje de máquinas y desarrolladores:

- **Ingeniero de datos *(Data Engineer):*** diseña, implementa y mantiene canalizaciones (pipelines) y sistemas de bases de datos para adquirir, limpiar y almacenar datos. Es el responsable de garantizar la calidad e integridad de los datos de forma automatizada con procesos de Extracción, Transformación y Carga (ETL) para mejorar la escalabilidad y el rendimiento de los sistemas de datos. Además, se responsabiliza por el versionado de datos.

- **Analista de datos *(Data Analyst):*** examina conjuntos de datos para identificar tendencias, patrones y relaciones que ayuden a tomar decisiones empresariales informadas. Es el responsable de presentar reportes sobre análisis exploratorios de datos con visualizaciones que facilitan la interpretación de la información y la extracción de conocimiento.

- **Científico de datos *(Data Scientist):*** aplica métodos científicos, procesos algorítmicos y sistemas para extraer conocimientos de datos estructurados y no estructurados. Utiliza técnicas de estadística, minería de datos, aprendizaje de máquinas y aprendizaje profundo para analizar e interpretar grandes conjuntos de datos. Es el responsable de resolver problemas complejos del negocio y apoyar la toma de decisiones basada en datos, combinando habilidades en programación, investigación y conocimiento del dominio específico del negocio. Igualmente, se responsabiliza de las etapas iniciales del ciclo de vida del modelo de ciencia

de datos desde la ingeniería de características, experimentación, entrenamiento, validación y versionado de modelos.

- **Ingeniero de Machine Learning *(Machine Learning Engineer):*** es el encargado de implementar y desplegar modelos de ciencia de datos en entornos de producción utilizando prácticas de desarrollo de integración y despliegue continuo con la infraestructura existente. Es el responsable del monitoreo y mantenimiento de los modelos y sus métricas en producción gestionando su ciclo de vida y aplicando actualizaciones según sea necesario; se responsabiliza por las pruebas unitarias de validación de modelos en los aspectos de rendimiento, precisión, sesgo y seguridad.

- **Desarrollador de aplicaciones *(Application Developer):*** se encarga del desarrollo y el mantenimiento de interfaces de usuario, APIs, y otros componentes de software necesarios para que los modelos de ciencia de datos puedan ser utilizados de manera efectiva en entornos de producción. Es el responsable de que las aplicaciones sean escalables, seguras y eficientes, trabajando en estrecha colaboración con ingenieros de machine learning y científicos de datos para implementar soluciones que cumplan con los requisitos del negocio; se responsabiliza por las pruebas unitarias de carga, integración, cobertura y usabilidad.

Rol de arquitectura de infraestructura de TI

- **Arquitecto de soluciones *(Solution Architect):*** es el encargado de diseñar e implementar la arquitectura global del sistema para asegurar la integración eficiente de modelos de ciencia de datos con aplicaciones empresariales. Es el responsable de administrar los recursos del sistema, ya sea en la nube, local o de forma híbrida. Además de asegurar la escalabilidad y seguridad, y la colaboración con equipos técnicos para garantizar la cohesión y la funcionalidad del sistema.

En el contexto de proyectos de ciencia de datos, se encuentra el rol de las partes interesadas solicitantes o *stakeholder,* el cual es una persona o grupo que tiene interés en el proyecto y proporciona requisitos estratégicos. Sus responsabilidades incluyen definir objetivos y expectativas, asegurar recursos, validar resultados y facilitar la comunicación entre el equipo de desarrollo y otros interesados. Su conocimiento del negocio y la visión estratégica garantizan que el proyecto se alinee con los objetivos organizacionales y genere valor tangible.

1.4. Herramientas y utilidades para proyectos de ciencia de datos

Herramientas y utilidades para proyectos de ciencia de datos

Las herramientas y utilidades para proyectos de ciencia de datos son aplicaciones y software diseñados para facilitar a un equipo de ciencia de datos ejecutar y optimizar proyectos de manera eficiente, asegurando precisión y escalabilidad.

En esta sección, las principales herramientas y utilidades empleadas en el ámbito empresarial diseñadas para facilitar y automatizar las diversas etapas del ciclo de vida de modelos en proyectos de ciencia de datos, desde su desarrollo inicial hasta su implementación y mantenimiento en producción, así como garantizar su reproducibilidad, se enumeran, según su propósito a continuación:

- **Gestión y almacenamiento de datos:** son sistemas diseñados para centralizar y organizar grandes volúmenes de información, facilitando su acceso y análisis en proyectos de ciencia de datos. Permiten la ingesta, el almacenamiento y la consulta de datos estructurados, no estructurados y semiestructurados.

- **Procesamiento de datos:** son sistemas diseñados para manipular y transformar grandes volúmenes de información. Permiten realizar desde análisis de datos exploratorios básicos hasta complejos análisis que involucran técnicas y métodos avanzados de ciencia de datos e inteligencia artificial. Tienen la capacidad para procesar datos en tiempo real o por lotes, dependiendo de las necesidades del proyecto.

- **Gestión de calidad y gobernanza de datos:** son componentes esenciales para garantizar confiabilidad en proyectos de ciencia de datos. Por un lado, las herramientas de gestión de calidad permiten detectar anomalías, perfilar, limpiar, enriquecer y monitorear datos, asegurando que los análisis y modelos de ciencia de datos se basen en información precisa y completa. Mientras que las herramientas de gobernanza de datos permiten tener control y trazabilidad

de acceso a los datos mediante políticas de acceso, uso, rastreo de origen y transformación de datos.

- **Desarrollo y entrenamiento de modelos:** son instrumentos que permiten la escritura y la ejecución de código, la manipulación y la visualización de datos, la creación de modelos, la evaluación del rendimiento y la integración con diferentes plataformas. Ofrecen entornos de desarrollo integrados, interfaces gráficas, soporte para diversos lenguajes de programación y bibliotecas especializadas en ciencia de datos.

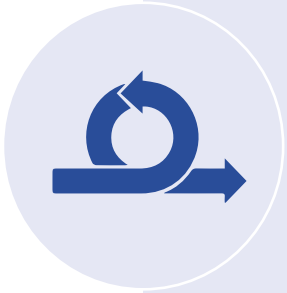
- **Herramientas de MLOps y CI/CD:** son herramientas de software esenciales para la automatización y gestión del ciclo de vida de modelos de ciencia de datos. Aseguran la reproducibilidad, escalabilidad y mantenimiento continuo de los modelos en producción. Permiten la automatización de las etapas de un proyecto, desde el desarrollo hasta la implementación, y ofrecen capacidades para la integración continua (CI), el despliegue continuo (CD), el versionado y documentación de código y el monitoreo en tiempo real. Facilitan la gestión de versiones, la integración y prueba automatizada de cambios, y el despliegue sin interrupciones de actualizaciones.

- **Visualización de datos:** son instrumentos para crear representaciones gráficas de datos complejos generalmente en tableros de control, facilitando el análisis y la interpretación de la información en proyectos de ciencia de datos. Transforman datos crudos en hallazgos *(insights)* visuales que permiten identificar patrones, tendencias y anomalías de manera eficiente.

- **Orquestación de Flujos de Trabajo:** son plataformas diseñadas para automatizar y gestionar la ejecución de tareas y procesos en proyectos de ciencia de datos. Aseguran la eficiencia y la reproducibilidad de los flujos de trabajo complejos, permitiendo una coordinación y monitoreo precisos de las tareas. Cuentan con la capacidad para manejar flujos de trabajo escalables, integrar múltiples fuentes de datos y herramientas de procesamiento, y la provisión de interfaces intuitivas para la visualización y control de los procesos.

- **Gestión de proyectos:** son plataformas diseñadas para facilitar la planificación, la ejecución y el seguimiento de actividades en proyectos. Ofrecen funciones avanzadas de seguimiento y monitoreo de actividades y entregables, utilizando metodologías para proporcionar una visualización clara del progreso del proyecto y la identificación de posibles desviaciones. Además, tienen la capacidad de ser compatibles con metodologías ágiles, permitiendo una gestión flexible y adaptativa de proyectos en entornos dinámicos.

1.5. Metodologías para la gestión de proyectos de ciencia de datos



Metodologías para la gestión de proyectos de ciencia de datos

Las metodologías para la gestión de proyectos de ciencia de datos son enfoques estructurados que guían la planificación, ejecución y control de proyectos asegurando un desarrollo eficiente, flexible y alineado con los objetivos del negocio.

Las principales metodologías para la gestión de proyectos de ciencia de datos utilizadas en el ámbito empresarial en proyectos de ciencia de datos se enumeran, según su estructura a continuación:

1.5.1. Metodologías secuenciales e iterativas

Las metodologías secuenciales e iterativas son un conjunto de enfoques tradicionales y estructurados para la ejecución de proyectos de ciencia de datos que, aunque son unidireccionales, lineales y planificados también son iterativas. Implican la división del proyecto en etapas claramente definidas, donde se debe completar cada etapa antes de pasar a la siguiente (Provost & Fawcett, 2013). A continuación, se listan las principales metodologías secuenciales utilizadas en la industria:

Knowledge Discovery in Databases (KDD)

Descubrimiento de Conocimiento en Bases de Datos (KDD) es una metodología en cascada, iterativa, incremental, unidireccional y que también puede implementarse de forma secuencial para la extracción de conocimiento útil y significativo a partir de grandes conjuntos de datos complejos. Se basa en la colaboración entre equipos multidisciplinarios y en la adaptación continua a medida que se avanza en el proceso de descubrimiento de conocimiento (Piatetsky-Shapiro, 1989). Se compone por cin-

co etapas: selección, preprocesamiento, transformación, minado de datos, interpretación y evaluación.

Sample, Explore, Modify, Model, Assess (SEMMA)

Es una metodología de minería de datos que consta de cinco pasos secuenciales e iterativos desarrollada por el Instituto SAS para resolver una amplia gama de problemas de análisis de datos de manera efectiva y eficiente (SAS Institute, 2005). Estos corresponden a: muestreo, exploración, modificación, modelamiento y evaluación.

Cross-Industry Standard Process for Data Mining (CRISP-DM)

Es una metodología estándar y altamente reconocida para minería de datos que consta de un proceso de seis fases interrelacionadas que describen el ciclo de vida típico de ciencia de datos (CRISP-DM Consortium, 1999). Estas son: entendimiento del negocio, entendimiento de los datos, preparación de datos, modelamiento, evaluación, despliegue y administración del modelo.

En la siguiente tabla se presenta la comparación de las etapas de las metodologías secuenciales en el campo de la ciencia de datos mencionadas previamente:

Tabla 1. Etapas de las metodologías secuenciales para ciencia de datos

Etapa	KDD	SEMMA	CRISP-DM
1	Pre KDD		Entendimiento del negocio
2	Selección	Muestreo	Entendimiento de los datos
3	Preprocesamiento	Exploración	
4	Transformación	Modificación	Preparación de los datos
5	Minería	Modelamiento	Modelamiento
6	Interpretación	Evaluación	Evaluación
7	Post KDD		Despliegue

Fuente: DANE, 2024, a partir de información de referentes internacionales.

1.5.2. Metodologías ágiles

Las metodologías ágiles son un conjunto de enfoques de desarrollo de software que se caracterizan por tener ciclos iterativos e incrementales cortos, generalmente de aproximadamente una semana, la colaboración estrecha entre equipos multidisciplinarios y la capacidad de adaptación a los cambios en los requisitos del cliente (Agile Alliance, 2001). En estas se resaltan los individuos y las interacciones sobre los procesos y las herramientas, el software funcionando sobre la documentación extensiva, la colaboración con el cliente sobre la negociación contractual y la respuesta ante el cambio sobre seguir un plan. Estas metodologías se fundamentan en SCRUM[2], que promueve la colaboración continua, la entrega incremental y la adaptabilidad, y en Kanban[3], que se enfoca en la optimización del flujo de trabajo y la mejora continua. Si bien fueron creadas inicialmente con un fin específico para el desarrollo de software, en la actualidad existen aplicaciones a ciencia de datos. Estas son las más utilizadas en la industria:

Agile Analytics

Es una metodología ágil que se centra en la entrega rápida de valor, la adaptabilidad a los cambios y la colaboración entre equipos de trabajo que aplica los principios y prácticas ágiles para el análisis de datos de manera iterativa e incremental, dividiendo el trabajo en pequeñas unidades de trabajo que pueden ser completadas en ciclos cortos de desarrollo (Collier, 2011). Es un enfoque facilitado, colaborativo, inclusivo y evolutivo para el modelado o la planificación que se organiza de acuerdo con las necesidades de los clientes.

Guerrilla Analytics

Es un enfoque de trabajo ágil y pragmático en el ámbito de la ciencia de datos que se enfoca en la aplicación de métodos y técnicas ágiles para resolver problemas de análisis de datos de manera eficiente priorizando la simplicidad y la eficacia sobre los procesos y las herramientas formales y pesados especialmente en entornos donde los recursos son limitados o las restricciones son significativas. Se compone por seis etapas: entendimiento del negocio, entendimiento de los datos, preparación de datos, modelamiento, evaluación y despliegue (Ridge, 2015).

Team Data Science Process (TDSP)

Es una metodología ágil desarrollada por Microsoft que busca estructurar el desarrollo de proyectos de ciencia de datos en varios niveles: ciclo de vida de ciencia de datos, estructuración del proyecto, infraestructura y recursos, herramientas y utilidades. Se compone por cuatro etapas: entendimiento del negocio, adquisición y entendimiento de los datos, modelamiento y despliegue. Además, se caracteriza por ser una metodo-

[2] SCRUM es un marco de trabajo ágil que facilita la colaboración en proyectos complejos, permitiendo la entrega incremental y rápida de valor (Schwaber & Sutherland, 2020). En SCRUM, los proyectos se dividen en iteraciones cortas llamadas sprints, donde los equipos multifuncionales trabajan en la implementación de funcionalidades priorizadas, siguiendo un ciclo de planificación, ejecución, revisión y retroalimentación.

[3] Kanban es un enfoque visual de gestión de flujo de trabajo que busca mejorar la eficiencia y la flexibilidad en la entrega deproductos o servicios (Anderson & Reinertsen, 2010). Se basa en la visualización del trabajo a través de un tablero de tareas, donde se limitan las tareas en progreso para optimizar el flujo y reducir cuellos de botella.

logía ágil que busca estructurar el desarrollo de proyectos de ciencia de datos en varios niveles: ciclo de vida de ciencia de datos, estructuración del proyecto, infraestructura y recursos y herramientas y utilidades (Microsoft, 2016).

Machine Learning Operations (MLOps)

Es una metodología ágil que combina prácticas de desarrollo de software (DevOps) con procesos específicos para el desarrollo, implementación y operación de sistemas de aprendizaje de máquinas (ML). Tiene como objetivo mejorar la colaboración entre equipos de trabajo, el desarrollo con la automatización de flujos de trabajo, la gestión del ciclo de vida de los modelos, el monitoreo del rendimiento y la escalabilidad de los modelos de ciencia de datos en entornos de producción (Machine Learning for Developers, 2022).

En la siguiente tabla se presenta la comparación de las etapas de las metodologías ágiles en el campo de la ciencia de datos:

Tabla 2. Etapas de las metodologías ágiles para ciencia de datos

Etapas	Agile Analytics	Guerrilla Analytics	TDSP	MLOps
1	Definición de objetivos	Entendimiento de negocio	Definición de objetivos	Planeación
2	Formación del equipo		Identificación de las fuentes de datos	Formulación
3	Recolección de datos	Extracción de datos	Ingesta de datos	Recolección

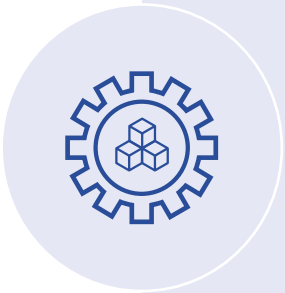
<table>
<tr><th>Etapas</th><th>Agile Analytics</th><th>Guerrilla Analytics</th><th>TDSP</th><th>MLOps</th></tr>
<tr><td>4</td><td>Limpieza de datos</td><td rowspan="4">Carga y recepción de datos</td><td rowspan="4">Preparación, exploración y limpieza de datos</td><td>Curación</td></tr>
<tr><td>5</td><td rowspan="4">Exploración de datos</td><td>Transformación</td></tr>
<tr><td>6</td><td>Validación</td></tr>
<tr><td>7</td><td rowspan="2">Exploración</td></tr>
<tr><td>8</td><td rowspan="3">Análisis y consolidación</td><td>Ingeniería de características</td></tr>
<tr><td>9</td><td>Selección y desarrollo de modelo</td><td>Entrenamiento del modelo</td><td>Entrenamiento</td></tr>
<tr><td>10</td><td>Evaluación del modelo</td><td>Evaluación del modelo</td><td>Evaluación</td></tr>
<tr><td>11</td><td colspan="3" rowspan="4"></td><td>Codificación</td></tr>
<tr><td>12</td><td>Construcción</td></tr>
<tr><td>13</td><td>Pruebas</td></tr>
<tr><td>14</td><td>Liberación</td></tr>
<tr><td>15</td><td>Despliegue</td><td rowspan="3">Productos del trabajo y reportes</td><td>Despliegue</td><td>Despliegue</td></tr>
<tr><td>16</td><td>Operación</td><td>Operacionalizar</td><td>Operación</td></tr>
<tr><td>17</td><td>Monitoreo</td><td>Monitoreo</td><td>Monitoreo</td></tr>
</table>

Fuente: DANE, 2024, a partir de información de referentes internacionales.

2.
INFRAESTRUCTURAS Y HERRAMIENTAS

En este capítulo se presentan los aspectos claves, beneficios y limitaciones de la infraestructura y herramientas utilizadas en el ámbito empresarial para gestionar las etapas del ciclo de vida de modelos en proyectos de ciencia de datos. Se expone un amplio espectro de herramientas categorizadas por sus propósitos específicos, con el fin de ofrecer al lector una visión completa del ámbito de la infraestructura de Tecnologías de la Información habilitada para proyectos de ciencia de datos. Lo anterior, como insumo para facilitar el diseño de arquitecturas de solución adaptadas al grado de dificultad del proyecto y los recursos asignados para su ejecución. Este capítulo se subdivide en siete subcapítulos: (2.1) infraestructura de Tecnologías de Información, (2.2) herramientas de gestión de paquetes y entornos virtuales, (2.3) control de versiones de código, (2.4) control de versiones de datos, (2.5) control de gestión de ciclo de vida de modelos, (2.6) distribución y ejecución de aplicaciones y (2.7) desarrollo, monitoreo y gestión de proyectos.

2.1. Infraestructura de Tecnologías de Información



Infraestructura de TI

Es el conjunto de componentes físicos y virtuales necesarios para el funcionamiento de los sistemas informáticos. Proporciona el soporte necesario para el almacenamiento, procesamiento, transmisión y protección de datos y servicios informáticos dentro de una organización.

La infraestructura de TI puede implementarse de diversas maneras, ya sea en las instalaciones locales de la empresa, a través de servicios en la nube ofrecidos por proveedores externos, o mediante una combinación de ambos enfoques, conocido como modelo híbrido. Cada uno de estos modelos presenta características únicas que impactan en la eficiencia operativa y la adaptabilidad a las necesidades del negocio y se describen a continuación:

Local: modelo de implementación de tecnología de la información en que los recursos informáticos, incluidos servidores, almacenamiento y software, se alojan y mantienen localmente en las instalaciones físicas de una organización.

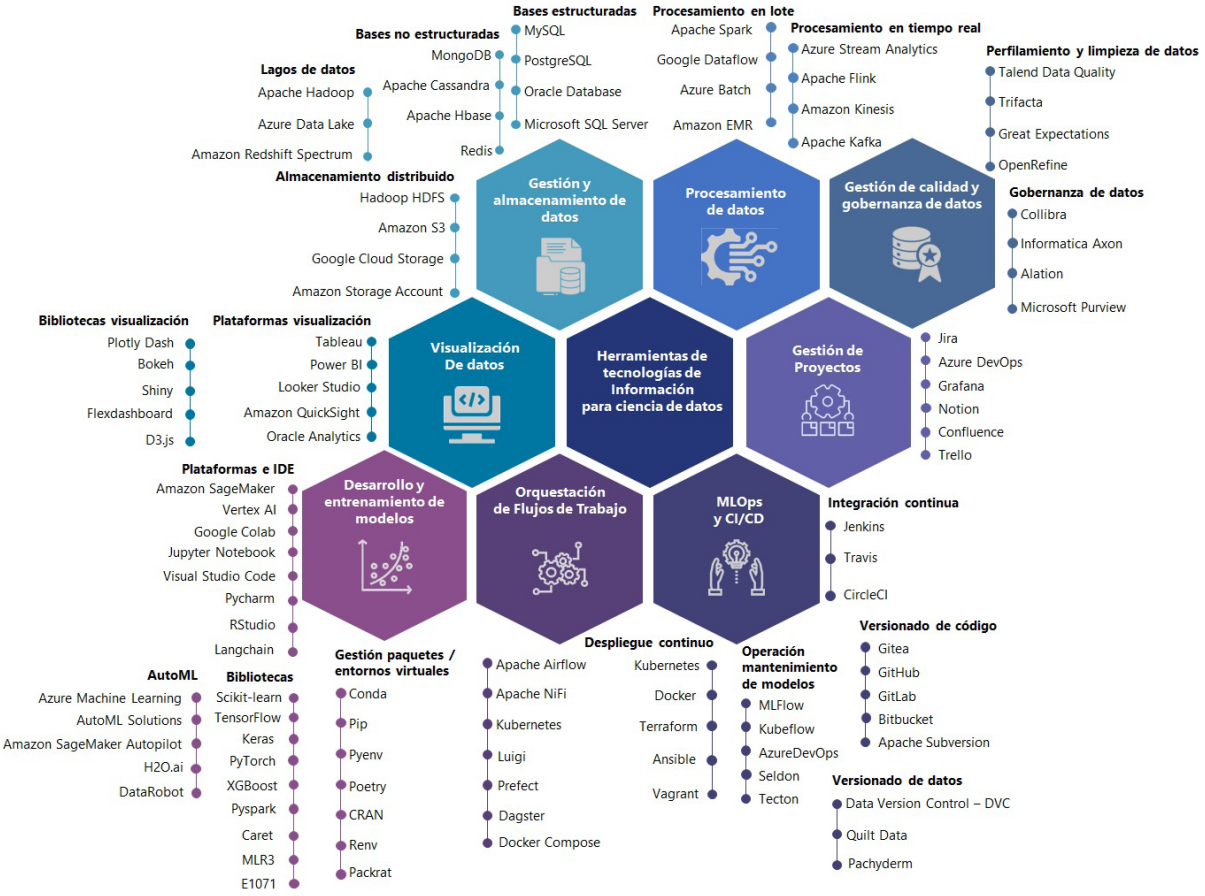
Nube: modelo de entrega de servicios de tecnología de la información a través de internet, donde los recursos informáticos, como servidores, almacenamiento, bases de datos, redes, software y otros servicios, se ofrecen y se acceden de manera remota a través de la red, generalmente a través de un proveedor de servicios en la nube.

Híbrido: modelo de implementación de tecnología de la información en el que una organización utiliza una combinación de recursos informáticos alojados localmente en sus instalaciones y recursos informáticos accesibles a través de internet a través de proveedores de servicios en la nube.

2.2. Herramientas de Tecnologías de Información

Las herramientas de TI son un conjunto de soluciones tecnológicas diseñadas para facilitar y optimizar diversas etapas del proceso de desarrollo y ejecución de proyectos de ciencia de datos. Garantizan la eficiencia, calidad y reproducibilidad de experimentos y análisis permitiendo a equipos de trabajo gestionar de manera efectiva los recursos, colaborar de manera transparente y mantener un control exhaustivo sobre los datos y modelos utilizados. Estas herramientas abarcan desde la gestión, almacenamiento, calidad, gobernanza, procesamiento y visualización de datos, hasta el control del ciclo de vida de modelos de ciencia de datos y la orquestación de flujos de trabajo. En definitiva, estas herramientas representan una pieza clave en la ejecución de proyectos de ciencia de datos, al proporcionar los recursos y procesos necesarios para gestionar de manera efectiva los datos, modelos y recursos computacionales involucrados en el ciclo de vida del proyecto. En la figura 1 se presentan un subconjunto de las principales herramientas de Tecnologías de Información para proyectos de ciencia de datos.

Figura 1. Herramientas de Tecnologías de Información para proyectos de ciencia de datos



Fuente: DANE, 2024.

2.2.1. Herramientas de gestión y almacenamiento de datos

Gestión y almacenamiento de datos

Diseñadas para almacenar, organizar y gestionar grandes volúmenes de datos tanto estructurados como no estructurados y semi estructurados. Permiten un acceso eficiente y seguro a los datos, facilitando su análisis y procesamiento.

Las herramientas de gestión y almacenamiento de datos son sistemas diseñados para centralizar y organizar grandes volúmenes de información, facilitando su acceso y análisis en proyectos de ciencia de datos. La centralización de los datos es esencial porque disminuye el movimiento de datos, reduciendo el riesgo de pérdida o corrupción. Además, facilita el trabajo colaborativo al proporcionar un punto de acceso único y consistente para todos los miembros de un equipo de trabajo. Estas herramientas permiten la ingesta, almacenamiento y consulta de datos estructurados, no estructurados y semiestructurados. Incluyendo la integración con diversas fuentes de datos y el manejo de grandes volúmenes de información.

A continuación, se presentan las herramientas más utilizadas en el ámbito empresarial, diferenciadas por el tipo de almacenamiento, junto con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

Bases de datos estructuradas

Las bases de datos estructuradas o relacionales son sistemas diseñados para organizar y gestionar datos de manera eficiente mediante un esquema predefinido basado en tablas y relaciones. Sus funciones principales incluyen la creación, actualización, consulta y administración de datos estructurados. Permiten manejar transacciones, asegurar la integridad referencial y optimizar consultas mediante índices. En definitiva, proporcionan una infraestructura robusta y confiable, esencial para el almacenamiento, gestión y consulta eficiente de datos estructurados.

A continuación, se presentan los gestores de bases de datos relacionales más utilizados en el ámbito empresarial, junto con sus características distintivas, que las hacen esenciales para el desarrollo de proyectos de ciencia de datos:

MySQL: desarrollado por MySQL AB y ahora propiedad de Oracle Corporation, es un sistema de gestión de bases de datos relacional (RDBMS) basado en SQL, que es conocido por su alta fiabilidad, escalabilidad y facilidad de uso. Sus principales funciones incluyen la gestión de bases de datos, la manipulación de datos y la administración de usuarios con control de acceso granular. Sin embargo, algunas limitaciones incluyen la falta de soporte completo para transacciones ACID y las capacidades limitadas para manejar cargas de trabajo muy grandes en comparación con otras bases de datos comerciales (Oracle Corporation, 2024a).

PostgreSQL: desarrollado por el grupo PostgreSQL Global Development Group, es un avanzado sistema de gestión de bases de datos relacional y objeto-relacional, conocido por su robustez, extensibilidad y conformidad con los estándares SQL. Sus principales funciones incluyen soporte para procedimientos almacenados, tipos de datos avanzados y control de concurrencia multi versión (MVCC). Algunas limitaciones incluyen una curva de aprendizaje pronunciada para nuevos usuarios y la complejidad en la configuración y optimización para obtener el máximo rendimiento (PostgreSQL Global Development Group, 2024).

Oracle Database: desarrollado por Oracle Corporation, es un sistema de gestión de bases de datos relacional (RDBMS) ampliamente utilizado, conocido por su alto rendimiento, escalabilidad y características de seguridad avanzadas. Sus principales funciones incluyen so-

porte para grandes volúmenes de datos, transacciones ACID completas, y características avanzadas de recuperación y replicación. No obstante, algunas de las limitaciones incluyen los altos costos de licenciamiento y la complejidad en su administración y configuración (Oracle Corporation, 2024c).

Microsoft SQL Server: Desarrollado por Microsoft, es un sistema de gestión de bases de datos relacional (RDBMS) que ofrece un amplio conjunto de herramientas para la gestión de datos, análisis e inteligencia de negocios. Sus principales funciones incluyen soporte para transacciones ACID, integración con servicios de análisis y reportes, y capacidades avanzadas de seguridad. Sin embargo, algunos desafíos incluyen los costos de licenciamiento elevados y la dependencia del entorno Windows para su implementación óptima (Microsoft Corporation, 2024b).

Cabe destacar que existen alternativas adicionales ampliamente utilizadas en la industria, como: MariaDB[4], IBM DB2[5] y SQLite[6].

Bases de datos no estructuradas

Estas son sistemas diseñados para gestionar y almacenar datos que no se ajustan a un esquema predefinido, como texto, imágenes y archivos multimedia. Permiten la flexibilidad y escalabilidad necesarias para manejar grandes volúmenes de datos diversos que no pueden ser organizados en un formato tabular tradicional. Sus funciones principales incluyen la captura, almacenamiento, indexación y consulta de datos no estructurados. Este tipo de bases de datos cuentan con la capacidad para manejar datos heterogéneos, escalabilidad horizontal y eficiencia en la búsqueda y recuperación de información.

A continuación, se enumeran las herramientas más empleadas en el entorno empresarial, destacando sus principales características que las distinguen en el desarrollo de proyectos de ciencia de datos:

MongoDB: desarrollado por MongoDB Incorporated, es una base de datos NoSQL orientada a documentos que almacena datos en un formato flexible, similar a JSON, permitiendo la fácil manipulación y consulta de datos complejos. Sus principales funciones incluyen escalabilidad horizontal, replicación, balanceo de carga y consultas ad hoc. Sin embargo, puede enfrentar desafíos en cuanto a la gestión de transacciones ACID completas y el rendimiento en operaciones de escritura intensivas en sistemas distribuidos (MongoDB Inc, 2024).

Apache Cassandra: desarrollada por Apache Software Foundation, es una base de datos NoSQL de columna amplia diseñada para manejar grandes volúmenes de datos distribuidos en múltiples servidores sin un único punto de fallo. Sus principales funciones incluyen alta disponibilidad, escalabilidad

[4] MariaDB es un sistema de gestión de bases de datos relacional de código abierto derivado de MySQL (MariaDB Foundation, 2024). Es conocido por su alto rendimiento, escalabilidad y compatibilidad con MySQL, ofreciendo mejoras adicionales en seguridad y rendimiento.
[5] IBM DB² es un sistema de gestión de bases de datos relacional desarrollado por (IBM, 2024b). Es utilizado en entornos empresariales para gestionar grandes volúmenes de datos y ofrece soporte para SQL, XML, y NoSQL, además de capacidades avanzadas de procesamiento transaccional.
[6] SQLite es un sistema de gestión de bases de datos relacional liviano, basado en archivos y de código abierto (SQLite, 2024). Es ampliamente utilizado en aplicaciones móviles, navegadores web y sistemas embebidos debido a su simplicidad y portabilidad.

horizontal y replicación automática de datos. No obstante, puede presentar limitaciones en la complejidad de las consultas y en la necesidad de una gestión cuidadosa de la configuración para optimizar el rendimiento (Apache Software Foundation, 2024g).

Apache Hbase: es una base de datos NoSQL de columna amplia diseñada para proporcionar almacenamiento y acceso en tiempo real a grandes tablas de datos dispersos. Sus principales funciones incluyen integración con Hadoop, escalabilidad horizontal y soporte para grandes volúmenes de datos. Sin embargo, puede enfrentar desafíos en la configuración y mantenimiento complejos, y en el manejo de consultas ad hoc debido a su enfoque en el acceso secuencial (Apache Software Foundation, 2024b).

Redis: desarrollado por Redis Labs, es una base de datos NoSQL en memoria de tipo clave-valor que ofrece almacenamiento rápido y persistente. Sus principales funciones incluyen soporte para diversas estructuras de datos, operaciones atómicas y capacidades de pub/sub. A pesar de sus ventajas, Redis puede enfrentar desafíos relacionados con la persistencia de datos en caso de fallos y la necesidad de memoria suficiente para manejar grandes conjuntos de datos en RAM (Redis Labs, 2024).

Cabe destacar que existen alternativas adicionales ampliamente utilizadas en la industria, como: Amazon Dynamo[7], Neo4J[8], Oracle Coherence[9], Couchbase[10], RavenDB[11], Cloudant[12] y Azure CosmosDB[13].

Almacenamiento distribuido

Los sistemas de almacenamiento distribuido son diseñados para almacenar y gestionar grandes volúmenes de datos distribuidos en múltiples nodos o servidores. Son fundamentales en entornos donde la escalabilidad, la disponibilidad y el rendimiento son críticos para el procesamiento eficiente de datos a gran escala. Permiten la partición y replicación de datos para garantizar la tolerancia a fallos y la capacidad de respuesta, así como la distribución de cargas de trabajo para un rendimiento óptimo. Tienen la capacidad para manejar datos semi estructurados y no estructurados proporcionando una alta disponibilidad y consistencia de datos en entornos distribuidos.

A continuación, se detallan las herramientas más populares en el ámbito empresarial, junto con las características clave que las hacen indispensables para proyectos de ciencia de datos:

Hadoop HDFS: desarrollado por Apache Software Foundation, es un sistema de archivos distribuido diseñado para almacenar

[7] Amazon DynamoDB es un servicio de base de datos NoSQL completamente administrado por (Amazon Web Services, 2024j), que ofrece almacenamiento rápido y flexible de documentos y datos clave-valor.
[8] Neo4j es una base de datos NoSQL orientada a grafos, diseñada para almacenar y gestionar relaciones complejas entre datos (Neo4j Inc, 2024).
[9] Oracle Coherence es una solución de almacenamiento distribuido en caché, utilizada para construir aplicaciones escalables y tolerantes a fallos (Oracle Corporation, 2024b).
[10] Couchbase es una base de datos NoSQL multimodelo que combina capacidades de base de datos de documentos y clave-valor (Couchbase Inc, 2024).
[11] RavenDB es una base de datos NoSQL orientada a documentos que ofrece almacenamiento transaccional, alta disponibilidad y replicación automática (RavenDB, 2024).
[12] IBM Cloudant es una base de datos NoSQL orientada a documentos y distribuida, basada en Apache CouchDB. Ofrece sincronización y acceso a datos a través de múltiples dispositivos y es utilizada para aplicaciones web y móviles (IBM, 2024a).
[13] Azure Cosmos DB es un servicio de base de datos NoSQL completamente administrado por Microsoft Azure, diseñado para proporcionar distribución global, escalabilidad horizontal y latencia baja con soporte para múltiples modelos de datos, incluyendo documentos, grafos y clave-valor (Microsoft Corporation, 2024a).

grandes cantidades de datos en clústeres de hardware de bajo costo. Sus principales funciones incluyen alta disponibilidad, tolerancia a fallos y escalabilidad horizontal mediante la replicación de datos. Sin embargo, puede enfrentar desafíos en la gestión de archivos pequeños y la complejidad en la configuración y el mantenimiento de los clústeres (Apache Software Foundation, 2024d).

Amazon S3: desarrollado por Amazon Web Services (AWS), es un servicio de almacenamiento de objetos escalable, duradero y seguro para cualquier tipo de datos. Sus principales funciones incluyen almacenamiento de objetos, control de versiones y políticas de ciclo de vida de los datos. No obstante, puede presentar desafíos relacionados con los costos de almacenamiento y transferencia de datos, así como las limitaciones en las capacidades de consulta directa sobre grandes volúmenes de datos (Amazon Web Services, 2024e).

Google Cloud Storage: desarrollado por Google, es un servicio de almacenamiento de objetos altamente escalable y duradero para datos de cualquier tipo y tamaño. Sus principales funciones incluyen almacenamiento de objetos, replicación automática y gestión de datos mediante políticas de ciclo de vida. Sin embargo, puede enfrentar desafíos en términos de costos asociados con el almacenamiento y la transferencia de datos, así como la necesidad de optimizar las configuraciones para un rendimiento óptimo (Google, 2024b).

Azure Storage Account: desarrollado por Microsoft, es un servicio de almacenamiento en la nube que proporciona almacenamiento escalable y duradero para objetos, archivos, colas y tablas. Sus principales funciones incluyen soporte para múltiples tipos de almacenamiento, alta disponibilidad y redundancia geográfica. Sin embargo, puede presentar desafíos en términos de costos de almacenamiento y transferencia de datos, así como la complejidad en la gestión y optimización de diferentes tipos de almacenamiento (Microsoft, 2024d).

Lagos de datos

Son sistemas diseñados para almacenar grandes volúmenes de datos de diversas fuentes y formatos en su forma original, sin necesidad de estructuración previa. Son fundamentales en entornos de ciencia de datos debido a su capacidad para gestionar datos heterogéneos y escalables, permitiendo un análisis exhaustivo y flexible. Sus funciones principales incluyen la captura, almacenamiento y procesamiento distribuido de datos, por lotes y en tiempo real. Tienen la capacidad para escalar horizontal y verticalmente, son compatibles con una amplia gama de herramientas de análisis y pueden integrarse con diversos sistemas de almacenamiento.

Se presenta a continuación las herramientas más utilizadas en el ámbito empresarial, con un enfoque en sus características principales que las destacan en el desarrollo de proyectos de ciencia de datos.

Apache Hadoop: desarrollado por Apache Software Foundation, es una plataforma de software de código abierto utilizada para la creación de lagos de datos que permite el almacenamiento y procesamiento distribuido de grandes conjuntos de datos utilizando el modelo de programación MapReduce. Sus principales funciones incluyen el almacenamiento distribuido a través de HDFS y el procesamiento de datos en paralelo utilizando MapReduce. Sin embargo, puede enfrentar desafíos en la complejidad de configuración y mantenimiento, así como en el manejo de cargas de trabajo en tiempo real debido a su naturaleza orientada a procesamiento por lotes (Apache Software Foundation, 2024d).

Azure Data Lake: desarrollado por Microsoft, es un servicio de almacenamiento y análisis de datos a escala masiva que permite la creación de lagos de datos para capturar y analizar cualquier tipo de datos. Sus principales funciones incluyen el almacenamiento de datos estructurados y no estructurados, la integración con herramientas de análisis y la escalabilidad masiva. Sin

embargo, los desafíos pueden incluir costos asociados con el almacenamiento y procesamiento de grandes volúmenes de datos, así como la necesidad de optimizar el rendimiento mediante configuraciones adecuadas (Microsoft, 2024i).

Amazon Redshift Spectrum: desarrollado por Amazon Web Services (AWS), es una extensión de Amazon Redshift para la creación de lagos de datos. Sus principales funciones incluyen la capacidad de consultar grandes volúmenes de datos almacenados en S3 y soporte para formatos de datos abiertos. Sin embargo, los desafíos pueden incluir costos de consulta y almacenamiento en S3, así como la necesidad de una gestión cuidadosa de los esquemas de datos para un rendimiento óptimo (Amazon Web Services, 2024d).

2.2.2. Herramientas de procesamiento de datos

Procesamiento de datos

Diseñadas para manejar y analizar grandes volúmenes de datos de manera eficiente. Permiten transformar y procesar datos masivos de manera rápida y flexible, optimizando flujos de trabajo y soportando una amplia gama de análisis, desde exploratorios hasta complejos análisis que involucran técnicas avanzadas de ciencia de datos e inteligencia artificial.

Las herramientas de procesamiento de datos son sistemas diseñados para manipular y transformar grandes volúmenes de información. Entre sus funciones principales se encuentran la limpieza, la transformación, el análisis y la visualización de datos. Permiten realizar desde análisis de datos exploratorios básicos hasta complejos análisis que involucran técnicas y métodos avanzados de ciencia de datos e inteligencia artificial. Son compatibles con una variedad de fuentes de datos y permiten manejar grandes volúmenes de información de manera eficiente. Tienen la capacidad para procesar datos en tiempo real o por lotes, dependiendo de las necesidades del proyecto.

A continuación, se presentan las herramientas más utilizadas en el ámbito empresarial, diferenciadas por el tipo de procesamiento, junto con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

Procesamiento en lote

Las herramientas de procesamiento de datos por lotes son componentes esenciales para el desarrollo de proyectos de ciencia de datos a gran escala, ya que permiten manipular y analizar grandes conjuntos de datos de manera eficiente. Se caracterizan por su escalabilidad, rentabilidad, paralelismo y flexibilidad, permitiendo manejar grandes volúmenes de información de diversos tipos, incluyendo datos estructurados, no estructurados y semi estructurados. El tipo de procesamiento que manejan permite optimizar la memoria y recursos al fragmentar los datos, reducir la carga de memoria activa, escalar horizontalmente y aprovechar algoritmos optimizados para análisis masivos.

A continuación, se enumeran las herramientas más empleadas en el entorno empresarial, destacando sus principales características que las distinguen en el desarrollo de proyectos de ciencia de datos:

Apache Spark: desarrollado por Apache Software Foundation, es un motor de análisis unificado para el procesamiento de grandes volúmenes de datos, conocido por su velocidad y facilidad de uso. Sus principales funciones incluyen procesamiento en memoria, soporte para múltiples lenguajes de programación y bibliotecas integradas para SQL, Machine Learning y procesamiento en lote y de gráficos. No obstante, algunos desafíos incluyen la gestión de memoria y la optimización de tareas para evitar cuellos de botella en sistemas distribuidos (Apache Software Foundation, 2024i).

Google Dataflow: desarrollado por Google, es un servicio totalmente gestionado para el procesamiento de datos en tiempo real y por lotes. Sus principales funciones incluyen la integración con el ecosistema de Google Cloud, la escalabilidad automática y la simplificación del desarrollo de pipelines de datos. Sin embargo, los desafíos pueden incluir la curva de aprendizaje inicial y los costos asociados con el procesamiento a gran escala y el uso de otros servicios de Google Cloud (Google, 2024c).

Azure Batch: desarrollado por Microsoft, es un servicio que permite ejecutar trabajos de computación paralela y por lotes a gran escala en la nube. Sus principales funciones incluyen la administración automática de recursos, la integración con otros servicios de Azure y la capacidad de ejecutar aplicaciones y scripts en paralelo. Sin embargo, algunos desafíos incluyen la configuración inicial y la gestión de costos en función del uso intensivo de recursos de computación (Microsoft, 2024h).

Amazon EMR: desarrollado por Amazon Web Services (AWS), es un servicio gestionado que facilita el procesamiento de grandes cantidades de datos gracias al modelo de programación Elastic Map Reduce mediante el uso de herramientas de código abierto como Apache Hadoop, Apache Spark, Apache HBase, Apache Flink y Presto. Sus principales funciones incluyen la rápida configuración de clústeres, el escalado automático de recursos y la integración con otros servicios de AWS. Sin embargo, puede enfrentar desafíos relacionados con la gestión de costos en entornos de procesamiento intensivo y la necesidad de una configuración cuidadosa para optimizar el rendimiento de los clústeres (Amazon Web Services, 2024a).

Procesamiento en tiempo real

Las herramientas de procesamiento de datos en tiempo real *(on-streaming)* son instrumentos fundamentales para el desarrollo de proyectos de ciencia de datos que requieren un análisis inmediato y oportuno de flujos de datos continuos. Se caracterizan por su baja latencia, alta escalabilidad y capacidad para manejar grandes volúmenes de información en tiempo real. El tipo de procesamiento que manejan optimiza el uso de recursos al procesar los datos a medida que llegan, evitando la necesidad de almacenar grandes volúmenes de datos en memoria y reduciendo el tiempo de procesamiento.

A continuación, se detallan las herramientas más populares en el ámbito empresarial, junto con las características clave que las hacen indispensables para proyectos de ciencia de datos:

Apache Kafka: desarrollado por Apache Software Foundation, es una plataforma de transmisión de datos distribuida que permite la construcción de pipelines de datos y aplicaciones de transmisión en tiempo real. Sus principales funciones incluyen la publicación y suscripción de flujos de registros, el almacenamiento de flujos de registros de manera tolerante a fallos y el procesamiento de flujos de registros en tiempo real. Sin embargo, puede enfrentar desafíos en la gestión de la complejidad operativa y la necesidad de recursos significativos para manejar grandes volúmenes de datos en tiempo real (Apache Software Foundation, 2024c).

Apache Flink: desarrollado por la Apache Software Foundation, es un motor de procesamiento de flujo y lote de datos que permite el procesamiento de eventos en tiempo real con baja latencia y alta tolerancia a fallos. Sus principales funciones incluyen el procesamiento de flujo en tiempo real, la capacidad de manejar grandes volúmenes de datos y el soporte para procesamiento de estado. No obstante, los desafíos pueden incluir la curva de aprendizaje empinada y la complejidad en la configuración y administración de los clústeres de Flink (Apache Software Foundation, 2024a).

Amazon Kinesis: desarrollado por Amazon Web Services (AWS), es una plataforma de transmisión en tiempo real que permite la recopilación, procesamiento y análisis de datos de transmisión a gran escala. Sus principales funciones incluyen la ingesta de grandes cantidades de datos en tiempo real, la capacidad de procesamiento y análisis en tiempo real y la integración con otros servicios de AWS. Sin embargo, puede enfrentar desafíos relacionados con los costos de procesamiento continuos y la necesidad de una configuración adecuada para optimizar el rendimiento (Amazon Web Services, 2024b).

Azure Stream Analytics: desarrollado por Microsoft, es un servicio de análisis en tiempo real totalmente gestionado que permite el procesamiento de datos de transmisión con consultas SQL simples. Sus principales funciones incluyen la ingesta y el procesamiento de datos en tiempo real, la integración con otros servicios de Azure y la capacidad de escalar automáticamente los recursos. Sin embargo, algunos desafíos pueden incluir la limitación en la complejidad de las consultas y los costos asociados con el procesamiento continuo de datos (Microsoft, 2024g).

2.2.3. Herramientas de gestión de calidad y gobernanza de datos

Gestión de calidad y gobernanza de datos

Diseñadas para asegurar la integridad, consistencia y confiabilidad de los datos. Permiten el perfilado, limpieza y validación de datos, así como la implementación de políticas y procedimientos para la gobernanza de estos.

Las herramientas de calidad de datos son componentes esenciales para garantizar confiabilidad en proyectos de ciencia de datos. Permiten detectar anomalías, perfilar, limpiar, enriquecer y monitorear datos, asegurando que los análisis y modelos de aprendizaje automático se basen en información precisa y completa. Generan un retorno significativo en términos de calidad, confiabilidad y valor de la información. Por otra parte, las herramientas de gobernanza permiten tener control y trazabilidad de acceso a los datos mediante políticas de acceso, uso, rastreo de origen y transformación de datos.

A continuación, se presentan las herramientas más utilizadas en el ámbito empresarial, diferenciadas por su propósito de calidad y gobernanza de datos, junto con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

Perfilamiento y limpieza de datos

Las herramientas de perfilamiento y limpieza de datos se utilizan para garantizar la calidad de los datos. Permiten analizar, identificar y corregir errores, inconsistencias, valores atípicos y duplicados en conjuntos de datos, asegurando que la información utilizada para análisis y modelado sea precisa, completa y confiable. Su uso optimiza el proceso de preparación de datos, reduce el tiempo y esfuerzo necesarios para obtener datos de calidad listos para su análisis.

A continuación, se presentan las herramientas más utilizadas en el ámbito empresarial, junto con sus características distintivas, que las hacen esenciales para el desarrollo de proyectos de ciencia de datos:

Talend Data Quality: desarrollado por Talend, es una solución que permite evaluar y mejorar la calidad de datos me-

diante herramientas avanzadas de perfilamiento, limpieza y enriquecimiento de datos. Sus principales funciones incluyen la identificación y corrección de datos duplicados, la estandarización de datos y la validación de datos contra reglas definidas por el usuario. Sin embargo, algunos desafíos pueden incluir la curva de aprendizaje asociada con la configuración de reglas complejas y la necesidad de integración con otros sistemas de Talend para obtener su máximo potencial (Talend, 2024).

Trifacta: creada por Trifacta Incorporation, es una plataforma de preparación y limpieza de datos que utiliza técnicas de aprendizaje de máquinas para automatizar y simplificar el perfilamiento, calidad y transformación de datos. Sus principales funciones incluyen la detección automática de esquemas, la sugerencia de transformaciones basadas en patrones de datos y la visualización interactiva de datos para facilitar su exploración y limpieza. No obstante, algunos desafíos pueden incluir los costos asociados con su uso a gran escala y la necesidad de formación para aprovechar plenamente sus capacidades avanzadas (Trifacta Incorporation, 2024).

Great Expectations: desarrollado por Superconductive, es una plataforma de validación de datos de código abierto que permite a las organizaciones definir, ejecutar y documentar pruebas de calidad de datos. Sus principales funciones incluyen la creación de suites de expectativas para validar datos, la integración con flujos de trabajo de datos y la generación de documentación de calidad de datos en formato legible por humanos. Sin embargo, los desafíos pueden incluir la complejidad en la configuración inicial y la necesidad de integración con otras herramientas de procesamiento de datos para ser completamente efectiva (Superconductive, 2024).

OpenRefine: es una herramienta de código abierto para la limpieza y transformación de datos. Sus principales funciones incluyen la edición masiva de datos, la transformación de datos utilizando expresiones regulares y el enlace de datos con bases de datos externas. No obstante, algunas limitaciones incluyen su enfoque en la manipulación de datos en memoria, lo que puede ser un desafío para conjuntos de datos extremadamente grandes, y la falta de integración directa con flujos de trabajo de datos automatizados (OpenRefine, 2024).

Gobernanza de datos

Las herramientas de gobernanza de datos son fundamentales para la gestión efectiva de datos en organizaciones. Permiten establecer políticas para el acceso, la creación, el almacenamiento, el uso y la eliminación de datos. Esto se traduce en optimización de la gestión de datos, la reducción de riesgos y el aumento de la confianza en los datos. Además, el linaje de datos permite tener transparencia y trazabilidad rastreando el origen, la transformación y el uso de los datos a lo largo de su ciclo de vida, proporcionando una visión completa del recorrido de la información.

A continuación, se presentan las herramientas más utilizadas en el ámbito empresarial con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

Collibra: desarrollada por Collibra Incorporation, es una plataforma de gobernanza de datos que permite gestionar y catalogar activos de datos, garantizando el cumplimiento normati-

vo y la calidad de los datos. Sus principales funciones incluyen la catalogación de datos, la gestión de políticas y la colaboración entre equipos para asegurar una toma de decisiones basada en datos confiables. Sin embargo, algunos desafíos pueden incluir la complejidad en la implementación inicial y la necesidad de formación especializada para aprovechar al máximo todas sus funcionalidades (Collibra Incorporation, 2024).

Informatica Axon: es una herramienta de gobernanza de datos que proporciona un enfoque colaborativo para la gestión de datos, permitiendo la creación y administración de políticas de datos, la catalogación de activos y el seguimiento del linaje de datos. Sus principales funciones incluyen la catalogación de datos, la gestión de calidad de datos y la colaboración entre equipos. Sin embargo, algunos desafíos pueden incluir los costos de licenciamiento y la integración con otros sistemas de TI existentes en la organización (Informatica, 2024).

Alation: desarrollada por Alation Incorporation, es una plataforma que combina la gobernanza de datos con la catalogación y el descubrimiento de datos, permitiendo a los usuarios buscar, analizar y gestionar datos de manera eficiente. Sus principales funciones incluyen la catalogación de datos, el linaje de datos, y la colaboración en torno a los datos. Sin embargo, algunos desafíos pueden incluir la necesidad de una integración robusta con otros sistemas de datos y la curva de aprendizaje asociada con la adopción de nuevas tecnologías en la organización (Sangani et al., 2024).

Microsoft Purview: desarrollada por Microsoft, es una herramienta de gobernanza de datos diseñada para proporcionar una vista unificada del panorama de datos de una organización. Sus principales funciones incluyen la catalogación automática de datos, el linaje de datos, y la clasificación de datos sensibles, facilitando así el descubrimiento y la gestión de datos en entornos híbridos. Sin embargo, algunos desafíos pueden incluir la necesidad de una integración robusta con otros sistemas de datos existentes y la curva de aprendizaje para los equipos que implementan nuevas tecnologías de gobernanza de datos (Microsoft, 2024e).

2.2.4. Herramientas de desarrollo y entrenamiento de modelos

Desarrollo y entrenamiento de modelos

Proporcionan entornos de desarrollo integrado, bibliotecas para crear, entrenar y evaluar modelos de ciencia de datos y gestión de paquetes y entornos virtuales. Facilitan el ciclo completo de desarrollo de modelos, desde la experimentación inicial hasta la optimización automatizada, permitiendo tanto la codificación manual como el uso de interfaces intuitivas de AutoML.

Las herramientas de desarrollo y entrenamiento de modelos son instrumentos esenciales para la ejecución de proyectos de analítica y ciencia de datos. Permiten la escritura de código, la manipulación y la visualización de datos, la creación de modelos, la evaluación del rendimiento y la integración con diferentes plataformas. Ofrecen entornos de desarrollo integrados, interfaces gráficas, soporte para diversos lenguajes de programación y bibliotecas especializadas en ciencia de datos. Posibilitan la creación de espacios aislados evitando conflictos de versiones de bibliotecas, simplifican la instalación, la actualización y la eliminación de bibliotecas y marcos de trabajo *(frameworks)* y aseguran la consistencia y la fiabilidad de resultados obtenidos en diferentes configuraciones de software.

A continuación, se presentan las herramientas más utilizadas en el ámbito empresarial, diferenciadas por entornos de desarrollo integrado, aprendizaje de máquinas automático y bibliotecas de ciencia de datos, junto con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

Plataformas y entornos de desarrollo integrado para ciencia de datos

Las plataformas y Entornos de Desarrollo Integrado (IDE) permiten escribir, depurar y ejecutar código de manera eficiente. Son fundamentales para agilizar el ciclo de análisis de datos y mejorar la precisión de los resultados. Entre sus funciones principales se encuentran la edición avanzada de scripts, la inte-

gración con bibliotecas y frameworks de ciencia de datos, y capacidades de visualización y depuración en tiempo real. Algunos son compatibles con múltiples lenguajes de programación como Python y R, tienen extensibilidad mediante plugins específicos para análisis de datos y disponibilidad de herramientas colaborativas que facilitan el trabajo en equipo.

A continuación, se describen las herramientas más utilizadas en el ámbito empresarial, con un enfoque en sus características distintivas que las hacen fundamentales para el desarrollo de proyectos de ciencia de datos:

Azure Databricks: es una plataforma de análisis unificada basada en Apache Spark que permite la colaboración entre científicos de datos, ingenieros de datos y analistas de negocio. Sus principales funciones incluyen la integración con Azure, el procesamiento de grandes volúmenes de datos en tiempo real y la ejecución de trabajos de ciencia de datos. Además, facilita el desarrollo de modelos predictivos y analíticos con un entorno optimizado para la codificación colaborativa en notebooks compartidos. Sin embargo, algunos desafíos pueden incluir la curva de aprendizaje para nuevos usuarios y los costos asociados con el uso intensivo de recursos en la nube (Microsoft, 2024a).

Amazon SageMaker: desarrollado por Amazon Web Services (AWS), es un servicio completamente gestionado que permite a desarrolladores y científicos de datos construir, entrenar y desplegar modelos de ciencia de datos a escala. Sus principales funciones incluyen herramientas integradas para la preparación de datos, algoritmos de entrenamiento optimizados y despliegue en producción. Sin embargo, los desafíos pueden incluir los costos asociados con el procesamiento y almacenamiento de datos y la necesidad de conocimientos especializados para optimizar los modelos (Amazon Web Services, 2024f).

Vertex AI: desarrollado por Google Cloud, es una plataforma unificada que permite a desarrolladores y científicos de datos construir, entrenar y desplegar modelos de ciencia de datos con herramientas integradas y automatizadas. Sus principales funciones incluyen la gestión de pipelines de ML, la opción de AutoML, la integración con otros servicios de Google Cloud y la optimización automática de modelos. Sin embargo, algunos desafíos pueden incluir la curva de aprendizaje inicial y los costos asociados con el uso continuo de los servicios en la nube (Google Cloud Platform, 2024).

Google Colab: desarrollado por Google, es un entorno de cuadernos Jupyter basado en la nube que permite a desarrolladores y científicos de datos escribir y ejecutar código Python y R en el navegador. Sus principales funciones incluyen el acceso a GPU y TPU gratuitas, la integración con Google Drive y la colaboración en tiempo real. Sin embargo, los desafíos pueden incluir limitaciones en el tiempo de ejecución y la necesidad de estar conectado a internet para acceder a las funcionalidades en la nube (Google, 2024d).

Jupyter Notebook: desarrollado por Project Jupyter, es una aplicación web de código abierto que permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. Sus principales funciones incluyen el soporte para múltiples lenguajes de programación, la integración con bibliotecas de datos y visualización, y la capacidad de ejecutar

código en bloques interactivos. Sin embargo, algunos desafíos pueden incluir la gestión de dependencias y versiones de paquetes, así como la configuración inicial para entornos personalizados (Project Jupyter, 2024).

Visual Studio Code: desarrollado por Microsoft, es un editor de código fuente ligero pero potente que soporta el desarrollo en múltiples lenguajes de programación, incluyendo Python, y ofrece herramientas integradas para la depuración y el control de versiones. Sus principales funciones incluyen la integración con Git, la depuración interactiva y una amplia gama de extensiones para mejorar la productividad. Sin embargo, los desafíos pueden incluir la necesidad de configurar manualmente entornos y extensiones para adaptarse a proyectos específicos (Microsoft, 2024f).

PyCharm: desarrollado por JetBrains, es un IDE específico para Python que ofrece herramientas avanzadas para la edición, depuración y pruebas de código. Sus principales funciones incluyen la integración con sistemas de control de versiones, soporte para frameworks web y de ciencia de datos, y una interfaz de usuario intuitiva. No obstante, los desafíos pueden incluir los costos de licenciamiento para la versión profesional y la necesidad de recursos significativos del sistema para proyectos grandes (JetBrains, 2024).

RStudio: desarrollado por RStudio, PBC, es un IDE para el lenguaje de programación R, diseñado para facilitar el desarrollo de análisis estadísticos y modelos de aprendizaje de máquinas. Sus principales funciones incluyen la edición de scripts R, la integración con paquetes de datos y visualización, y herramientas para la depuración y el control de versiones. Sin embargo, los desafíos pueden incluir la curva de aprendizaje para nuevos usuarios y la necesidad de gestionar manualmente dependencias y paquetes (Posit PBC, 2024a).

Langchain: desarrollado por Harrison Chase, es una plataforma y entorno de desarrollo integrado (IDE) diseñado para construir aplicaciones de inteligencia artificial que aprovechan modelos de lenguaje. Facilita la creación de flujos de trabajo complejos mediante la integración con múltiples herramientas y modelos, incluyendo LLMs y bases de datos, proporcionando un marco para la orquestación de tareas y el manejo de cadenas de prompts. Sus principales funciones incluyen la fácil composición de cadenas de procesamiento, la integración con APIs externas y la personalización de pipelines de datos. Sin embargo, los desafíos pueden incluir una curva de aprendizaje empinada y la necesidad de configurar adecuadamente los modelos y las herramientas externas para lograr un rendimiento óptimo (LangChain, 2024).

AutoML

Las herramientas de AutoML *(Automated Machine Learning)* son plataformas que automatizan el proceso de creación de modelos de aprendizaje automático, permitiendo a el desarrollo y despliegue de modelos de manera más rápida y eficiente. Sus funciones principales incluyen la selección automática de características, la optimización de hiper parámetros, y la validación y evaluación de modelos. Generalmente destacan por su facilidad de uso, capacidad de integrar y preprocesar grandes volúmenes de datos, y la compatibilidad con múltiples algoritmos y frameworks de ciencia de datos.

A continuación, se presentan las herramientas más utilizadas en el ámbito empresarial con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

Azure Machine Learning: desarrollado por Microsoft, es un servicio en la nube que permite construir, entrenar e implementar modelos de ciencia de datos, incluyendo capacidades de AutoML. Sus principales funciones incluyen la automatización del entrenamiento de modelos, la gestión de experimentos, y la implementación y monitoreo de modelos en producción. Sin embargo, algunos desafíos pueden incluir los costos asociados con el uso extensivo de recursos en la nube y la necesidad de conocimientos técnicos para configurar y optimizar los modelos (Microsoft, 2024j).

AutoML Solutions: desarrollado por Google, es una suite de herramientas dentro de Vertex AI que permite entrenar modelos de ciencia de datos personalizados con facilidad y sin necesidad de experiencia avanzada en ML. Sus principales funciones incluyen la capacidad de entrenar modelos de visión, lenguaje, traducción y tabulares automáticamente. Sin embargo, los desafíos pueden incluir la curva de aprendizaje para nuevos usuarios y los costos asociados con el uso de servicios de Google Cloud (Google, 2024a).

Amazon SageMaker Autopilot: desarrollado por AWS, es un servicio que automatiza la construcción, el entrenamiento y el ajuste de modelos de ciencia de datos. Sus principales funciones incluyen la generación automática de pipelines de ML, la selección y ajuste de modelos, y la posibilidad de ver y modificar cada paso del proceso automático. Sin embargo, los desafíos pueden incluir los costos de procesamiento continuo y la necesidad de comprender los aspectos técnicos subyacentes para ajustar manualmente los modelos generados (Amazon Web Services, 2024g).

H2O.ai: desarrollada por H2O.ai Incorporated, es una plataforma de código abierto que proporciona herramientas de AutoML para la automatización del proceso de construcción de modelos de ciencia de datos. Sus principales funciones incluyen la automatización del entrenamiento y ajuste de modelos, el soporte para múltiples algoritmos de ML y la integración con entornos de datos en la nube y locales. No obstante, los desafíos pueden incluir la complejidad de configuración y la necesidad de conocimientos técnicos para optimizar los modelos generados (H2O.ai, 2024).

DataRobot: desarrollado por DataRobot Incorporated, es una plataforma de machine learning automatizada que permite a los usuarios construir y desplegar modelos ciencia de datos a gran escala sin necesidad de experiencia avanzada en ML. Sus principales funciones incluyen la automatización de todo el ciclo de vida de modelos de ciencia de datos, la explicación de modelos y la implementación de modelos en producción. Sin embargo, los desafíos pueden incluir los costos de licenciamiento y la necesidad de formación para aprovechar plenamente todas sus capacidades avanzadas (DataRobot, 2024).

Bibliotecas de ciencia de datos

Son herramientas fundamentales que facilitan el desarrollo y la implementación de modelos de ciencia de datos. Permiten el preprocesamiento de datos, la construcción de modelos, la evaluación de rendimiento y la optimización de hiper parámetros. Son compatibles con múltiples lenguajes de programación,

cuentan con una extensa disponibilidad de documentación y comunidades activas, y tienen la capacidad de integrarse con otras herramientas y plataformas de ciencia de datos. Además, facilitan la creación de modelos desde cero, lo que resulta primordial para la personalización de estos de acuerdo con los objetivos específicos de cada proyecto.

A continuación, se detallan las herramientas más utilizadas en el ámbito empresarial con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

Scikit-learn: desarrollado por el equipo de Scikit-learn y disponible como proyecto de código abierto, es una biblioteca de machine learning en Python que proporciona herramientas simples y eficientes para el análisis predictivo de datos. Sus principales funciones incluyen algoritmos de clasificación, regresión, clustering y reducción de dimensionalidad. Sin embargo, algunos desafíos pueden incluir la gestión de grandes volúmenes de datos en memoria y la necesidad de preprocesamiento extensivo de datos antes de su uso (Cournapeau et al., 2024).

TensorFlow: desarrollado por Google Brain Team, es una biblioteca de código abierto para la computación numérica y machine learning a gran escala, conocida por su flexibilidad y capacidad de despliegue en diferentes plataformas. Sus principales funciones incluyen soporte para redes neuronales profundas, modelos de aprendizaje supervisado y no supervisado, y una API flexible para construir y entrenar modelos de ML. Sin embargo, los desafíos pueden incluir la curva de aprendizaje empinada y la complejidad en la configuración de modelos avanzados (Google Brain Team, 2024).

Keras: inicialmente desarrollado por François Chollet y ahora parte del proyecto TensorFlow, es una biblioteca de redes neuronales de alto nivel en Python que facilita la construcción y el entrenamiento de modelos de aprendizaje profundo. Sus principales funciones incluyen una interfaz simple y consistente, compatibilidad con múltiples entornos de ejecución *(backends)* de aprendizaje profundo, y herramientas para el modelado rápido de prototipos. No obstante, los desafíos pueden incluir limitaciones en el control detallado de modelos avanzados y dependencias con backends como TensorFlow (Chollet, 2015).

PyTorch: desarrollado por Meta AI, es una biblioteca de ciencia de datos de código abierto que proporciona una plataforma flexible para el desarrollo e implementación de modelos de aprendizaje profundo. Sus principales funciones incluyen la construcción dinámica de gráficos computacionales, soporte para entrenamiento distribuido y herramientas de depuración avanzadas. Sin embargo, algunos desafíos pueden incluir la gestión de recursos computacionales y la necesidad de optimización manual para modelos complejos (Meta AI, 2016).

XGBoost: desarrollado por Tianqi Chen y ahora mantenido por una comunidad de código abierto, es una biblioteca de optimización de gradiente para problemas de clasificación y regresión que es conocida por su eficiencia y rendimiento. Sus principales funciones incluyen soporte para árboles de decisión en boosting, regularización L1 y L2, y manejo de datos faltantes. No obstante, los desafíos pueden incluir la configuración de hiper parámetros complejos y la necesidad de un entendimiento profundo de los algoritmos de impulse para su uso óptimo (Chen, 2023).

Pyspark: desarrollado como parte de Apache Spark, es la API de Python para Spark que permite realizar análisis de datos a gran escala y de machine learning en clústeres distribuidos. Sus principales funciones incluyen el procesamiento distribuido de datos, integración con MLlib para machine learning, y capacidad de manejo de grandes conjuntos de datos. Sin embargo, algunos desafíos pueden incluir la configuración inicial de clústeres y la optimización de tareas distribuidas (Apache Software Foundation, 2024e).

Caret: desarrollado por Max Kuhn, es un paquete de R que simplifica el proceso de creación de modelos predictivos, proporcionando herramientas para preprocesamiento de datos, selección de modelos y validación cruzada. Sus principales funciones incluyen una interfaz unificada para múltiples algoritmos de ciencia de datos, herramientas para la selección de características y evaluación de modelos. Sin embargo, los desafíos pueden incluir la gestión de dependencias de R y la necesidad de recursos computacionales significativos para modelos grandes (Kuhn, 2024).

MLR3: desarrollado por el equipo de MLR en R, es una biblioteca de *machine learning* moderna y flexible que proporciona una infraestructura para el aprendizaje supervisado y no supervisado en R. Sus principales funciones incluyen soporte para múltiples algoritmos de ML, herramientas para la evaluación y la comparación de modelos, y una arquitectura modular. Sin embargo, los desafíos pueden incluir la complejidad en la personalización de flujos de trabajo y la curva de aprendizaje para nuevos usuarios (Lang, 2024).

E1071: desarrollado por el Departamento de Estadística de la Universidad Técnica de Viena, es un paquete de R que incluye funciones para el aprendizaje estadístico y el análisis predictivo, especialmente conocido por su implementación de máquinas de vectores de soporte. Sus principales funciones incluyen algoritmos de clasificación, regresión y clustering, y herramientas para el análisis exploratorio de datos. Sin embargo, algunos desafíos pueden incluir limitaciones en la documentación y la necesidad de conocimientos avanzados en estadística y machine learning (Meyer, 2024).

Adicionalmente, existen APIs[14] que permiten el acceso directo a modelos de inteligencia artificial en la nube, como las ofrecidas por OpenAI[15] y Hugging Face[16]. Estas facilitan el uso de modelos pre entrenados en procesamiento de lenguaje natural, visión por computadora, generación de texto, análisis de sentimientos, traducción automática, entre otros, sin la necesidad de infraestructura local.

[14] Una API (Application Programming Interface) es un conjunto de definiciones y protocolos que permite que diferentes aplicaciones de software se comuniquen entre sí. Una API define las reglas y formatos a seguir para que un sistema o aplicación exponga sus funcionalidades a otros sistemas, facilitando la interacción entre distintas aplicaciones de forma estandarizada.
[15] La OpenAI API es un servicio en la nube que permite el acceso directo a modelos avanzados de inteligencia artificial, como GPT-4 (OpenAI, 2024). Ofrece capacidades para procesamiento de lenguaje natural, generación de texto, análisis de sentimientos, y más, facilitando la integración de estas tecnologías en diversas aplicaciones mediante una interfaz RESTful.
[16] Hugging Face Inference API proporciona acceso directo a una vasta colección de modelos de inteligencia artificial preentrenados en la nube, abarcando tareas como procesamiento de lenguaje natural, visión por computadora, y más. Esta API permite la integración de modelos de manera rápida y sencilla en aplicaciones y servicios (Hugging Face, 2024).

Gestión de paquetes y entornos virtuales

Las herramientas de gestión de paquetes y entornos virtuales facilitan la administración de las bibliotecas y las dependencias específicas requeridas para la reproducibilidad de resultados en proyectos de ciencia de datos. Simplifican la instalación, la actualización y la eliminación de bibliotecas y marcos de trabajo *(frameworks)*. Posibilitan la creación de espacios aislados evitando conflictos de versiones, asegurando la consistencia y la fiabilidad de los resultados obtenidos en diferentes configuraciones de software. Esto es especialmente valioso para garantizar la replicabilidad de experimentos y colaborar de manera efectiva entre equipos.

A continuación, se presentan las herramientas más utilizadas en el ámbito empresarial con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

Conda: es una herramienta de gestión de paquetes y entornos virtuales, desarrollada por Travis Oliphant, principalmente para Python y R, que permite la instalación, actualización y gestión de paquetes y sus dependencias de forma eficiente y reproducible. Es de código abierto y está licenciada bajo la licencia BSD. Sus principales funciones incluyen la creación de entornos virtuales independientes, la gestión de dependencias y la distribución de paquetes. Sin embargo, Los principales desafíos de esta herramienta son la gestión de dependencias complejas en múltiples lenguajes y plataformas, lo que puede ser especialmente difícil al trabajar en proyectos que requieren diferentes configuraciones de entorno (Oliphant, 2012).

Pip: es un sistema de gestión de paquetes de Python de código abierto desarrollado por Ian Bicking que facilita la instalación y gestión de paquetes y sus dependencias desde el Python Package Index (PyPI). Permite la instalación, actualización y eliminación de paquetes de forma sencilla y también es compatible con la instalación de paquetes desde archivos comprimidos. No obstante, sus principales desafíos son los conflictos de versiones y dependencias no resueltas durante la instalación y actualización de paquetes, lo que complica el mantenimiento del entorno de desarrollo (Bicking, 2008).

Pyenv: es una herramienta de gestión de entornos virtuales para Python desarrollado contributivamente que permite a los usuarios instalar, gestionar y cambiar fácilmente entre múltiples versiones de Python en un mismo sistema. Es de código abierto y sus funciones principales incluyen la gestión de versiones de Python y la creación de entornos virtuales independientes. Respecto a sus principales desafíos destacan la administración dificultosa de múltiples versiones de Python, particularmente cuando se requiere compatibilidad específica para diferentes proyectos y librerías (Pyenv Contributors, 2024).

Poetry: es una herramienta de gestión de dependencias y empaquetado para proyectos de Python desarrollada por Sébastien Eustace que simplifica el manejo de dependencias y la distribución de paquetes. Permite la instalación y gestión de dependencias de manera declarativa a través de un archivo pyproject.toml y facilita la creación y publicación de paquetes en PyPI. Sin embargo, sus principales desafíos son la curva de aprendizaje pronunciada y los problemas de compatibilidad con otros gestores de paquetes, a pesar de su capacidad para

simplificar la gestión de dependencias y versiones (Eustace, 2024).

Renv: es una herramienta de gestión de entornos virtuales para R que permite la creación y la gestión de entornos de R independientes. Permite la instalación de paquetes de manera aislada en cada entorno virtual, lo que facilita la reproducibilidad y la colaboración en proyectos de R. No obstante, Sus principales desafíos son la sincronización de dependencias exactas en diferentes sistemas operativos y configuraciones al gestionar entornos de R (Renv Contributors, 2024).

Packrat: es una herramienta de gestión de dependencias para R que facilita la creación de entornos virtuales reproducibles al encapsular las dependencias de un proyecto de R en una carpeta local. Permite la instalación y la gestión de paquetes de manera independiente para cada proyecto, lo que garantiza la coherencia y la reproducibilidad de los entornos. Sin embargo, sus principales desafíos son la sobrecarga de mantenimiento debido a la necesidad de controlar las versiones exactas de cada paquete, lo que puede ser laborioso y propenso a errores al mantener proyectos de R (RStudio, 2024a).

CRAN: la Red Integral de Archivos de R (Comprehensive R Archive Network – CRAN), es la red oficial de repositorios de paquetes para el lenguaje de programación R. Permite la distribución y gestión de bibliotecas necesarias para el desarrollo en este lenguaje. Facilita la instalación de paquetes y proporciona un marco estándar para la creación y el mantenimiento de entornos reproducibles al garantizar que los usuarios accedan a versiones compatibles de los paquetes. Sin embargo, uno de sus principales desafíos es la falta de control granular sobre las versiones de paquetes en diferentes proyectos, lo que puede llevar a inconsistencias si se trabaja en múltiples entornos o con dependencias complejas (R Core Team, 2024).

2.2.5. Herramientas de MLOps y CI/CD

MLOps y CI/CD

Diseñadas para automatizar, gestionar y monitorear el ciclo de vida de modelos en proyectos de ciencia de datos, garantizando la reproducibilidad, escalabilidad y mantenimiento continuo, a la vez que facilitan una entrega ágil y confiable de soluciones de ciencia de datos.

Las herramientas de Machine Learning Operations (MLOps) y de Integración Continua y Despliegue Continuo (CI/CD) son herramientas de software esenciales para la automatización y gestión del ciclo de vida de modelos de ciencia de datos. Son cruciales porque aseguran la reproducibilidad, escalabilidad y mantenimiento continuo de los modelos en producción. Permiten la gestión automatizada de versiones, la implementación sin interrupciones y el seguimiento en tiempo real del rendimiento del modelo.

A continuación, se presentan las herramientas más utilizadas en el ámbito empresarial, diferenciadas por CI, CD y MLOps, junto con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos.

Integración continua

Las herramientas de Integración Continua (CI) son plataformas diseñadas para automatizar la integración del código en proyectos de ciencia de datos, facilitando el desarrollo colaborativo y la detección temprana de errores. Son esenciales porque aseguran la calidad y consistencia del código al permitir pruebas y validaciones automáticas con cada cambio realizado. Sus funciones principales incluyen la compilación automatizada, la ejecución de pruebas unitarias y la generación de informes. Son compatibles con múltiples sistemas de control de versiones, permiten la configuración flexible de canalizaciones y la integración con otras herramientas de desarrollo y despliegue.

A continuación, se detallan las herramientas más utilizadas en el ámbito empresarial con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

Jenkins: desarrollado originalmente por Kohsuke Kawaguchi y ahora mantenido por la comunidad de código abierto bajo la supervisión de la Jenkins Project, es una plataforma de integración continua de código abierto que facilita la automatización del proceso de desarrollo de software. Sus principales funciones incluyen la automatización de la construcción y prueba de aplicaciones, la integración con numerosos plugins y herramientas de terceros, y la capacidad de desplegar automáticamente aplicaciones. Sin embargo, algunos desafíos pueden incluir la complejidad en la configuración inicial y el mantenimiento de plugins, así como la gestión de recursos en grandes entornos de desarrollo (Kawaguchi, 2024).

Travis: desarrollado por Travis CI, es una plataforma de integración continua que se integra directamente con GitHub, permitiendo la automatización de pruebas y despliegues de software. Sus principales funciones incluyen la automatización de la construcción y la prueba de proyectos, el soporte para múltiples lenguajes de programación y la capacidad de desplegar aplicaciones en diversas plataformas de hospedaje. No obstante, los desafíos pueden incluir limitaciones en la personalización avanzada y los costos asociados con los planes premium para proyectos privados y de gran escala (Travis CI, 2024).

CircleCI: desarrollado por Circle Internet Services, es una plataforma de integración y entrega continuas que permite a los desarrolladores automatizar la construcción, prueba y despliegue de su código. Sus principales funciones incluyen la configuración basada en YAML, la integración con repositorios de código como GitHub y Bitbucket, y la capacidad de ejecutar pipelines en entornos de contenedores o máquinas virtuales.

Sin embargo, los desafíos pueden incluir la curva de aprendizaje inicial para configurar pipelines complejos y los costos asociados con los planes avanzados para proyectos empresariales (CircleCI, 2024).

Cabe destacar que existen alternativas adicionales ampliamente utilizadas en la industria, como AWS codepipeline[17], Azure pipelines[18] y Atlassian Bamboo[19].

Despliegue continuo

Las herramientas de Despliegue Continuo (CD) estan diseñadas para automatizar el proceso de implementación de modelos y aplicaciones en proyectos de ciencia de datos, garantizando una entrega rápida y fiable. Resultan cruciales porque permiten la actualización y escalabilidad continua de los entornos de producción, minimizando el tiempo de inactividad y los errores manuales. Sus funciones principales incluyen la orquestación de contenedores, la gestión de infraestructura como código y la automatización de configuraciones. Permiten la integración con sistemas de CI, el manejo de múltiples entornos de despliegue y la facilidad de escalar aplicaciones de manera dinámica.

A continuación, se presentan las herramientas más utilizadas en el ámbito empresarial con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

Kubernetes: desarrollado por Google y ahora mantenido por la Cloud Native Computing Foundation (CNCF), es una plataforma de orquestación de contenedores de código abierto diseñada para automatizar la implementación, escalado y gestión de aplicaciones en contenedores, facilitando su despliegue continuo. Sus principales funciones incluyen la gestión de clústeres de contenedores, el balanceo de carga y la recuperación automática de fallos. Sin embargo, algunos desafíos pueden incluir la complejidad en la configuración inicial y la gestión operativa, así como la curva de aprendizaje empinada para administradores y desarrolladores (Cloud Native Computing Foundation, 2024).

Docker: desarrollado por Salomon Hykes, es una plataforma de contenerización que permite a los desarrolladores empaquetar aplicaciones y sus dependencias en contenedores portátiles y consistentes, simplificando el despliegue continuo. Sus principales funciones incluyen la creación y gestión de contenedores, la integración con herramientas de CI/CD y la facilidad de despliegue en diferentes entornos. Sin embargo, algunos desafíos pueden incluir la gestión de datos persistentes y la seguridad de los contenedores, así como la necesidad de conocimientos avanzados para optimizar el rendimiento de los contenedores (Docker Incorporated, 2024).

[17] AWS CodePipeline es un servicio de entrega continua que automatiza las etapas de construcción, prueba y despliegue de aplicaciones en la nube de (Amazon Web Services, 2024h).
[18] Azure Pipelines es un servicio de integración y entrega continua (CI/CD) en la nube de (Microsoft, 2024c). Soporta múltiples lenguajes de programación y entornos, permitiendo la automatización de procesos de construcción, prueba y despliegue para cualquier aplicación, en cualquier plataforma.
[19] Atlassian Bamboo es una herramienta de integración y entrega continuas que permite la automatización de la construcción, prueba y despliegue de software. Ofrece integración profunda con otras herramientas de Atlassian, como Jira y Bitbucket, para una gestión de proyectos unificada (Atlassian, 2024a).

Terraform: desarrollado por HashiCorp, es una herramienta de infraestructura como código que permite definir y proporcionar infraestructura a través de archivos de configuración declarativos, apoyando el despliegue continuo de infraestructuras complejas. Sus principales funciones incluyen la gestión de infraestructura de múltiples proveedores, el uso de un lenguaje de configuración declarativo y la capacidad de realizar un seguimiento de los cambios de infraestructura. No obstante, los desafíos pueden incluir la complejidad en la escritura y mantenimiento de archivos de configuración y la necesidad de integración con otros sistemas de gestión de infraestructura (HashiCorp, 2024a).

Ansible: desarrollado por Ansible Community y Red Had Incorporated, es una herramienta de automatización de código abierto que permite la configuración, gestión y despliegue de aplicaciones e infraestructura a través de archivos de configuración YAML, facilitando el despliegue continuo. Sus principales funciones incluyen la automatización de tareas de TI, la gestión de configuración y la orquestación de despliegues. Sin embargo, algunos desafíos pueden incluir la gestión de grandes volúmenes de inventarios y la necesidad de conocimientos específicos para la creación de playbooks y roles efectivos (Ansible Community & Red Hat, 2024).

Vagrant: desarrollado por HashiCorp, es una herramienta de código abierto que facilita la creación y gestión de entornos de desarrollo virtualizados. Utilizando archivos de configuración descriptivos (Vagrantfiles), permite automatizar la provisión y la configuración de máquinas virtuales, asegurando que los entornos de desarrollo sean consistentes y reproducibles. Sus principales funciones incluyen la gestión de entornos de desarrollo en múltiples plataformas y la integración con herramientas de virtualización como VirtualBox y Docker. Sin embargo, algunos desafíos pueden incluir la sobrecarga en recursos de hardware y la curva de aprendizaje asociada con la configuración avanzada de Vagrantfiles (HashiCorp, 2024b).

Operación y mantenimiento de modelos de ciencia de datos

Las herramientas de operación y mantenimiento de modelos (MLOps) son soluciones diseñadas para supervisar el rendimiento y asegurar la integridad continua de modelos de ciencia de datos en producción. Son fundamentales porque permiten la detección temprana de desviaciones en el rendimiento y aseguran que los modelos sigan siendo precisos y relevantes con el tiempo. Sus funciones principales incluyen el seguimiento de métricas, la gestión de versiones de modelos y la automatización del reentrenamiento. Tienen la capacidad de integrarse con canalizaciones de CI/CD, permiten la implementación en entornos de producción y la provisión de alertas en tiempo real para cambios significativos en el rendimiento.

A continuación, se detallan las herramientas más utilizadas en el ámbito empresarial con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

MLflow: desarrollado por el equipo de Matei Zaharia, es una plataforma de código abierto para gestionar el ciclo de vida de los modelos de machine learning, incluyendo experimentación, reproducibilidad y despliegue. Sus principales funciones incluyen el seguimiento de experimentos, la gestión de modelos y el despliegue de modelos. Sin embargo, algunos desafíos pueden incluir la

necesidad de integrar MLflow con otras herramientas y plataformas de ML y la complejidad de configuración inicial para entornos a gran escala (MLflow incorporated, 2024).

Kubeflow: desarrollado por Google y la comunidad de código abierto, es una plataforma de código abierto diseñada para facilitar el despliegue, operación y escalado de flujos de trabajo de machine learning en Kubernetes. Sus principales funciones incluyen la gestión de pipelines de ML, el entrenamiento distribuido de modelos y el soporte para múltiples marcos de trabajo *(frameworks)* de ML. Sin embargo, los desafíos pueden incluir la complejidad en la configuración y gestión de clústeres de Kubernetes y la curva de aprendizaje empinada para nuevos usuarios (Google, 2024f).

Azure DevOps: desarrollado por Microsoft, es una plataforma que ofrece un conjunto de herramientas para gestionar el ciclo de vida del desarrollo de software, incluyendo la integración continua, entrega continua y la gestión de versiones. Sus principales funciones incluyen la integración de repositorios de código, la automatización de pipelines de CI/CD y la colaboración entre equipos. Sin embargo, algunos desafíos pueden incluir la necesidad de configuración avanzada para proyectos complejos y los costos asociados con los planes de uso a gran escala.

Seldon: desarrollado por Seldon Technologies, es una plataforma de código abierto para el despliegue, escalado y gestión de modelos de machine learning en Kubernetes. Sus principales funciones incluyen la orquestación de modelos en contenedores, el monitoreo del rendimiento de los modelos y la gestión de versiones de modelos. Sin embargo, los desafíos pueden incluir la integración con sistemas existentes y la necesidad de conocimientos en Kubernetes para una implementación efectiva (Seldon Technologies, 2024).

Tecton: desarrollado por Tecton.ai, es una plataforma de gestión de características que facilita la creación, el mantenimiento y la operación de datos de características para modelos de machine learning. Sus principales funciones incluyen la ingeniería de características, la gestión de versiones de características y la integración con pipelines de ML. Sin embargo, algunos desafíos pueden incluir los costos de implementación y la necesidad de una integración profunda con los sistemas de datos y ML existentes (Tecton.AI, 2024).

Versionado de código

Las herramientas de versionado de código están diseñadas para gestionar y controlar las versiones del código fuente, documentación o cualquier artefacto generado por un proceso de desarrollo de software facilitando el desarrollo colaborativo y el seguimiento de cambios. Son esenciales porque aseguran la integridad y la trazabilidad del código, permitiendo revertir cambios y colaborar eficientemente entre múltiples desarrolladores. Sus funciones principales incluyen la gestión de ramas, la fusión de cambios, y el seguimiento de historial de versiones. Tienen la capacidad de: integrarse con sistemas de CI/CD, colaboración y revisión de código, y el soporte para repositorios distribuidos. En resumen, las herramientas de versionado de código son fundamentales para mantener la coherencia y la calidad en el desarrollo de proyectos de ciencia de datos, permitiendo un control preciso y una colaboración efectiva en equipos de desarrollo.

A continuación, se presentan las herramientas más utilizadas en el ámbito empresarial con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

Gitea: desarrollado por la comunidad de código abierto y gestionado por la organización Gitea, es una plataforma de desarrollo colaborativo auto hospedada y ligera, basada en Git. Sus principales funciones incluyen la gestión de repositorios Git, la colaboración en código y la integración con CI/CD. Sin embargo, algunos desafíos pueden incluir la necesidad de infraestructura propia para el alojamiento y la configuración y mantenimiento manuales del servidor (Gitea Community, 2024).

GitHub: desarrollado por GitHub Incorporated y ahora parte de Microsoft, es una plataforma de desarrollo colaborativo basada en Git que ofrece herramientas para la gestión de código, la integración continua y la revisión de código. Sus principales funciones incluyen la creación y gestión de repositorios, la colaboración en proyectos y la integración con numerosas herramientas de desarrollo y CI/CD. Sin embargo, los desafíos pueden incluir los costos asociados con los planes avanzados y las limitaciones de privacidad en los repositorios públicos (Github Incorporated, 2024b).

GitLab: desarrollado por GitLab Incorporated, es una plataforma completa de DevOps que proporciona herramientas para la gestión de repositorios de código, CI/CD, y la planificación y monitorización de proyectos. Sus principales funciones incluyen la gestión de código fuente, la automatización de pipelines de CI/CD y la integración de la gestión de proyectos y la seguridad. Sin embargo, los desafíos pueden incluir la complejidad en la configuración inicial y los costos de licenciamiento para características avanzadas (Gitlab incorporated, 2024a).

Bitbucket: desarrollado por Atlassian, es un servicio de alojamiento de repositorios Git y Mercurial que facilita la colaboración en código y la integración continua. Sus principales funciones incluyen la gestión de repositorios, la colaboración y la integración con otras herramientas de Atlassian como Jira. No obstante, los desafíos pueden incluir la curva de aprendizaje para nuevos usuarios y las limitaciones de funciones en los planes gratuitos (Atlassian, 2024b).

Apache Subversion – SVN: desarrollado por la CollabNet, es un sistema de control de versiones de código abierto que proporciona gestión de versiones centralizada. Sus principales funciones incluyen la gestión de versiones, el control de acceso y la capacidad de manejar grandes archivos y directorios. Sin embargo, algunos desafíos pueden incluir la falta de soporte nativo para operaciones distribuidas y la necesidad de configuración y mantenimiento del servidor centralizado (CollabNet, 2024).

Versionado de datos

Las herramientas de versionado de datos estan diseñadas para gestionar y controlar las versiones de conjuntos de datos utilizados en proyectos de ciencia de datos, asegurando la reproducibilidad y la integridad de los análisis. Son esenciales porque permiten rastrear cambios en los datos, facilitando la colaboración y el cumplimiento de normas y regulaciones. Sus funciones principales incluyen el seguimiento de versiones de conjuntos de datos, la integración con flujos de trabajo de ciencia de datos

y la auditoría de cambios. Cuentan con la capacidad de integrarse con sistemas de control de versiones de código y de provisión de un historial detallado de modificaciones en los datos.

A continuación, se presentan las herramientas más utilizadas en el ámbito empresarial con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

Data Version Control – DVC: desarrollado por Iterative, es una herramienta de control de versiones de datos diseñada para gestionar datos de proyectos de ciencia de datos y flujos de trabajo basados en Git. Sus principales funciones incluyen la gestión de versiones de datos y modelos, la reproducción de experimentos y la integración con sistemas de control de versiones de código como Git. Sin embargo, algunos desafíos pueden incluir la curva de aprendizaje inicial para nuevos usuarios y la necesidad de integración con otras herramientas de machine learning y almacenamiento de datos (Iterative.ai, 2024).

Quilt Data: desarrollado por Quilt Data Incorporated, es una plataforma de control de versiones de datos que facilita la gestión, el almacenamiento y la colaboración en conjuntos de datos. Sus principales funciones incluyen la creación de versiones de conjuntos de datos, la integración con S3 para el almacenamiento de datos y la colaboración en equipos a través de catálogos de datos. Sin embargo, algunos desafíos pueden incluir los costos asociados con el almacenamiento en la nube y la necesidad de formación para aprovechar completamente sus capacidades (Quilt Data Incorporated, 2024).

Pachyderm: desarrollado por Pachyderm Incorporated, es una plataforma de datos que combina el control de versiones de datos con el procesamiento distribuido en contenedores. Sus principales funciones incluyen la gestión de versiones de datos, la integración con Kubernetes para el procesamiento de datos y el soporte para pipelines reproducibles. Sin embargo, los desafíos pueden incluir la complejidad en la configuración inicial y la necesidad de conocimientos en Kubernetes y contenedores para su implementación efectiva (Pachyderm Incorporated, 2024).

Cabe destacar que existen alternativas adicionales ampliamente utilizadas en la industria, como GIT LFS[20], LakeFS[21] y Dolt[22].

[20] Git LFS es una extensión del sistema de control de versiones Git diseñada para manejar grandes archivos y datos binarios de forma eficiente (Github Incorporated, 2024a).
[21] LakeFS es una plataforma de control de versiones para datos almacenados en lagos de datos, que permite gestionar cambios en grandes volúmenes de datos de manera similar a Git (Treeverse Inc, 2024).
[22] Dolt es una base de datos SQL que incluye capacidades de control de versiones, permitiendo realizar commits, ramas y fusiones directamente en los datos (DoltHub, 2024).

2.2.6. Herramientas de visualización de Datos

Visualización de Datos

Diseñadas para transformar datos complejos en representaciones gráficas intuitivas generalmente en forma de tableros de control que facilitan el análisis, la comunicación e interpretación de resultados.

Las herramientas de visualización de datos se utilizan para crear representaciones gráficas de datos complejos, facilitando el análisis y la interpretación de la información en proyectos de ciencia de datos. Son esenciales porque transforman datos crudos en hallazgos *(insights)* visuales que permiten identificar patrones, tendencias y anomalías de manera eficiente. Sus funciones principales incluyen la creación de tableros de control interactivos, la integración con diversas fuentes de datos y la personalización de visualizaciones. Tienen interfaces gráficas intuitivas que facilitan su uso para usuarios no técnicos, y permiten la integración con otras herramientas de análisis y almacenamiento de datos.

A continuación, se presentan las herramientas más utilizadas en el ámbito empresarial, diferenciadas por plataformas, softwares y bibliotecas especializadas en el diseño de tableros de control, junto con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos.

Plataformas para visualización de datos

Las plataformas para visualización de datos están diseñadas específicamente para facilitar la implementación rápida y sencilla de tableros de control. Permiten la integración con múltiples fuentes de datos y cuentan con un catálogo extenso de tipos de visualizaciones, controles y filtros para transformar datos en hallazgos visuales en cuestión de pocos clics. Destacan por su facilidad de uso para usuarios no técnicos.

A continuación, se detallan las herramientas más utilizadas en el ámbito empresarial con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

Tableau: desarrollado por Salesforce, es una plataforma de visualización de da-

tos que permite a los usuarios crear gráficos interactivos y dashboards a partir de una amplia gama de fuentes de datos. Sus principales funciones incluyen la conexión a múltiples fuentes de datos, la creación de visualizaciones interactivas y la capacidad de compartir tableros de control en línea. Sin embargo, algunos desafíos pueden incluir los costos de licenciamiento elevados y la curva de aprendizaje inicial para dominar sus funcionalidades avanzadas (Salesforce, 2024).

Power BI: desarrollado por Microsoft, es una herramienta de análisis empresarial que permite a los usuarios conectar, visualizar y compartir datos de manera interactiva. Sus principales funciones incluyen la integración con diversas fuentes de datos, la creación de informes y tableros de control interactivos y la capacidad de publicar y compartir informes a través de la nube. Sin embargo, algunos desafíos pueden incluir la complejidad de configuración inicial y los costos asociados con las versiones premium (Microsoft, 2024k).

Looker Studio: desarrollado por Google, es una plataforma de inteligencia empresarial y visualización de datos que permite a los usuarios explorar, analizar y compartir insights de datos de manera eficiente. Sus principales funciones incluyen la integración con múltiples fuentes de datos, la creación de tableros de control interactivos y el soporte para análisis ad-hoc. Sin embargo, los desafíos pueden incluir los costos de licenciamiento y la necesidad de conocimientos técnicos para aprovechar completamente sus capacidades (Google, 2024e).

Amazon QuickSight: desarrollado por Amazon Web Services (AWS), es un servicio de inteligencia de negocio que permite a los usuarios crear y publicar tableros de control interactivos y análisis visuales de datos. Sus principales funciones incluyen la integración con diversas fuentes de datos, la creación de análisis interactivos y el uso de machine learning para insights avanzados. Sin embargo, algunos desafíos pueden incluir los costos variables en función del uso y la necesidad de integración con otros servicios de AWS para obtener el máximo rendimiento (Amazon Web Services, 2024c).

Oracle Analytics: es una plataforma de análisis de datos que proporciona herramientas para la visualización de datos, el análisis predictivo y la creación de informes interactivos. Sus principales funciones incluyen la integración con una amplia gama de fuentes de datos, la capacidad de crear tableros de control interactivos y el uso de inteligencia artificial para análisis avanzados. Sin embargo, los desafíos pueden incluir la complejidad de configuración inicial y los costos de licenciamiento (Oracle, 2024).

Bibliotecas para visualización de datos

Estas bibliotecas están diseñadas para crear gráficos y visualizaciones a partir de datos, facilitando el análisis y la interpretación en proyectos de ciencia de datos. Son esenciales porque permiten presentar resultados de manera clara y efectiva, ayudando en la identificación de patrones, tendencias y anomalías. Sus funciones principales incluyen la generación de gráficos estáticos y dinámicos, la personalización de visualizaciones y la integración con otros paquetes de análisis de datos. Tienen la capacidad de flexibilidad y personalización para adaptarse a diversas necesidades de visualización, además de compatibi-

lidad con múltiples lenguajes de programación y entornos.

A continuación, se presentan las herramientas más utilizadas en el ámbito empresarial con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

Plotly Dash: desarrollado por Plotly, es una biblioteca de Python que facilita la creación de aplicaciones web analíticas interactivas sin necesidad de conocimientos avanzados en desarrollo web. Sus principales funciones incluyen la capacidad de crear tableros de control interactivos, la integración con bibliotecas de visualización de datos como Plotly y la compatibilidad con múltiples lenguajes de programación. Sin embargo, algunos desafíos pueden incluir la curva de aprendizaje inicial para usuarios nuevos y la necesidad de optimización para aplicaciones de gran escala (Plotly, 2024).

Bokeh: desarrollado por Anaconda Inc., es una biblioteca de visualización interactiva en Python que permite crear gráficos interactivos y tableros de control web escalables. Sus principales funciones incluyen la creación de visualizaciones interactivas, la integración con otras bibliotecas de datos de Python y el soporte para despliegues en aplicaciones web. Sin embargo, algunos desafíos pueden incluir la complejidad en la creación de visualizaciones altamente personalizadas y la necesidad de recursos computacionales significativos para visualizaciones muy grandes (Bokeh Contributors, 2024).

Shiny: desarrollado por Posit PBC, es una biblioteca de R que facilita la construcción de aplicaciones web interactivas directamente desde código R y Python, permitiendo la visualización de datos en tiempo real. Sus principales funciones incluyen la capacidad de crear tableros de control interactivos, la integración con paquetes de visualización de datos de R y Python y la facilidad de despliegue en servidores web. Sin embargo, algunos desafíos pueden incluir la escalabilidad para aplicaciones grandes y la necesidad de conocimientos avanzados en R para maximizar su potencial (Posit PBC, 2024b).

Flexdashboard: desarrollado por RStudio, es una biblioteca de R que permite la creación de tableros de control interactivos con un enfoque en la simplicidad y la facilidad de uso. Sus principales funciones incluyen la creación de tableros de control con lenguaje de marcado *(markdown),* la integración con gráficos R y la capacidad de publicar en múltiples formatos. Sin embargo, los desafíos pueden incluir limitaciones en la personalización avanzada y la dependencia de otros paquetes de R para funcionalidades más complejas (RStudio, 2024b).

D3.js: desarrollada por Mike Bostock, es una biblioteca de JavaScript ampliamente utilizada para la visualización de datos dinámicos y basados en web. Permite la manipulación directa de documentos basados en datos, utilizando estándares web como SVG, HTML y CSS, lo que proporciona un control granular sobre el diseño y la animación de visualizaciones. Sus principales funciones incluyen la creación de gráficos interactivos, la integración con datos en tiempo real y la capacidad de construir visualizaciones altamente personalizadas. Sin embargo, los desafíos pueden incluir una curva de aprendizaje empinada y la necesidad de un conocimiento profundo de JavaScript y estándares web para aprovechar plenamente sus capacidades (Bostock, 2024).

2.2.7. Herramientas de orquestación de Flujos de Trabajo

Orquestación de Flujos de Trabajo

Permiten automatizar, coordinar, optimizar y supervisar la ejecución de tareas en flujos de trabajo complejos, asegurando la eficiencia y la reproducibilidad de las operaciones. Gestionan dependencias, programan tareas y proporcionan una visibilidad integral del estado de los flujos de trabajo, lo que facilita la detección de errores y la mejora continua de procesos.

Las herramientas de orquestación de flujos de trabajo están diseñadas para automatizar y gestionar la ejecución de tareas y procesos en proyectos de ciencia de datos. Resultan esenciales porque aseguran la eficiencia y la reproducibilidad de los flujos de trabajo complejos, permitiendo una coordinación y monitoreo precisos de las tareas. Sus funciones principales incluyen la programación y la ejecución de tareas, la gestión de dependencias y la supervisión de los procesos en tiempo real. Cuentan con la capacidad para manejar flujos de trabajo escalables, integrar múltiples fuentes de datos y herramientas de procesamiento, y la provisión de interfaces intuitivas para la visualización y el control de los procesos.

A continuación, se presentan las herramientas más utilizadas en el ámbito empresarial con sus características principales que las distinguen en el desarrollo de proyectos de ciencia de datos:

Apache Airflow: desarrollado por Apache Software Foundation, es una plataforma de código abierto para la creación, programación y monitoreo de flujos de trabajo programados. Sus principales funciones incluyen la definición de flujos de trabajo como gráficos acíclicos dirigidos (DAG), la capacidad de escalar a miles de tareas y la integración con una variedad de servicios y APIs. Sin embargo, algunos desafíos pueden incluir la complejidad en la configuración inicial y la necesidad de gestionar dependencias manualmente (Apache Software Foundation, 2024f).

Apache NiFi: desarrollado por Apache Software Foundation, es una herramienta de integración de datos en tiempo real que permite el diseño y gestión de flujos

de datos de manera visual. Sus principales funciones incluyen la automatización del movimiento y transformación de datos entre sistemas, la gestión de flujos de trabajo en tiempo real y la capacidad de rastrear y visualizar el flujo de datos. Sin embargo, algunos desafíos pueden incluir la complejidad en la configuración de flujos de datos complejos y la necesidad de recursos significativos para operaciones intensivas (Apache Software Foundation, 2024h).

Kubernetes: desarrollado por Google y ahora mantenido por la Cloud Native Computing Foundation (CNCF), es una plataforma de orquestación de contenedores de código abierto que automatiza el despliegue, escalado y operación de aplicaciones en contenedores. Sus principales funciones incluyen la gestión de clústeres de contenedores, el balanceo de carga y la recuperación automática de fallos. Sin embargo, los desafíos pueden incluir la complejidad en la configuración inicial y la gestión operativa, así como la curva de aprendizaje empinada para administradores y desarrolladores (Cloud Native Computing Foundation, 2024).

Luigi: desarrollado por Spotify, es una herramienta de código abierto para la creación de pipelines de procesamiento de datos complejos que se ejecutan en lotes. Sus principales funciones incluyen la definición de tareas y dependencias, la gestión de la ejecución de tareas y la reejecución de tareas fallidas. Sin embargo, algunos desafíos pueden incluir la escalabilidad limitada para flujos de trabajo extremadamente grandes y la necesidad de configuración manual para tareas complejas (Spotify, 2024).

Prefect: desarrollado por Prefect Technologies Inc., es una plataforma de orquestación de flujos de trabajo que facilita la programación y monitorización de flujos de trabajo de datos y machine learning. Sus principales funciones incluyen la creación y la gestión de flujos de trabajo mediante un lenguaje declarativo, la integración con múltiples servicios y la capacidad de manejar flujos de trabajo dinámicos. Sin embargo, los desafíos pueden incluir la necesidad de una infraestructura de backend para la gestión y la posible complejidad en la configuración inicial para usuarios nuevos (Prefect Technologies, 2024).

Dagster: desarrollado por Elementl, es una plataforma de orquestación de datos que proporciona un entorno para la creación y ejecución de pipelines de datos con un enfoque en la reutilización de código y la modularidad. Sus principales funciones incluyen la definición de pipelines como gráficos de operaciones, la integración con herramientas de datos y la capacidad de depuración y pruebas integradas. Sin embargo, algunos desafíos pueden incluir la curva de aprendizaje inicial y la necesidad de integración con sistemas existentes para maximizar su uso (Elementl, 2024).

Docker Compose: desarrollado por Docker Inc, es una herramienta de código abierto que facilita la orquestación de aplicaciones multi-contenedor, permitiendo la definición y gestión de servicios a través de un archivo YAML. Sus principales funciones incluyen la configuración de redes, volúmenes y dependencias entre contenedores, así como el despliegue y escalado de aplicaciones en un solo comando. Sin embargo, los desafíos pueden incluir la falta de soporte para la orquestación a gran escala y la necesidad de integrar herramientas adicionales como Kubernetes para manejar entornos de producción complejos (Docker Incorporated, 2024).

2.2.8. Herramientas de gestión de proyectos

Gestión de proyectos

Facilitan el desarrollo, el monitoreo y la gestión de proyectos teniendo como principal enfoque la colaboración de equipos de trabajo y el seguimiento y el análisis de las etapas del ciclo de vida de un proyecto.

Las herramientas de desarrollo, monitoreo y gestión de proyectos son plataformas diseñadas para facilitar la planificación, ejecución y seguimiento de actividades en proyectos. Su importancia radica en su capacidad para optimizar la colaboración, mejorar la eficiencia y garantizar la entrega exitosa de los proyectos dentro del plazo y presupuesto establecidos. Ofrecen funciones avanzadas de seguimiento y monitoreo de actividades y entregables, utilizando metodologías para proporcionar una visualización clara del progreso del proyecto y la identificación de posibles desviaciones. Además, tienen la capacidad de ser compatibles con metodologías ágiles, permitiendo una gestión flexible y adaptativa de proyectos en entornos dinámicos.

Se detalla a continuación una lista de las herramientas más comúnmente empleadas en el ámbito empresarial, enfocándose en sus características distintivas que las identifican como elementos fundamentales en el desarrollo de proyectos de ciencia de datos:

Jira: desarrollado por Atlassian, es una herramienta de gestión de proyectos diseñada para la planificación, seguimiento y gestión de proyectos de software. Sus principales funciones incluyen la gestión de proyectos ágil mediante tableros Kanban y Scrum, la creación y el seguimiento de incidencias, y la generación de informes detallados. Sin embargo, algunos desafíos pueden incluir la complejidad en la configuración inicial y los costos de licenciamiento para equipos grandes (Atlassian, 2024d).

Azure DevOps: desarrollado por Microsoft, es una suite de herramientas que permite la planificación, desarrollo y entrega de software de manera continua. Sus principales funciones incluyen la integración y entrega continuas (CI/CD), la gestión de repositorios de código fuente, y la planificación ágil de proyectos. Sin embargo, los desafíos pue-

den incluir la curva de aprendizaje para nuevos usuarios y los costos asociados con las funcionalidades avanzadas (Microsoft, 2024b).

Grafana: desarrollada por Grafana Labs, es una plataforma de código abierto para la analítica y el monitoreo que permite visualizar métricas y logs de diferentes fuentes de datos. Sus principales funciones incluyen la creación de tableros de control interactivos, la integración con múltiples fuentes de datos y la configuración de alertas en tiempo real. Sin embargo, algunos desafíos pueden incluir la necesidad de conocimientos técnicos para configurar fuentes de datos y la gestión de grandes volúmenes de datos en tiempo real (Grafana Labs, 2024).

Notion: desarrollada por Notion Labs Inc., es una herramienta de productividad que combina notas, bases de datos, tareas y wikis en un solo espacio de trabajo colaborativo. Sus principales funciones incluyen la creación de documentos y bases de datos, la organización de proyectos y tareas, y la colaboración en tiempo real. Sin embargo, los desafíos pueden incluir la complejidad en la organización de grandes volúmenes de contenido y los costos asociados con los planes premium (Notion Labs, 2024).

Confluence: desarrollada por Atlassian, es una plataforma de colaboración en equipo que permite crear, compartir y organizar el contenido del proyecto. Sus principales funciones incluyen la creación de documentos y wikis, la integración con Jira y otras herramientas de Atlassian, y la gestión de conocimiento del equipo. Sin embargo, algunos desafíos pueden incluir los costos de licenciamiento y la complejidad en la organización de contenido en equipos grandes (Atlassian, 2024c).

Trello: desarrollada por Atlassian, es una herramienta de gestión de proyectos basada en tableros visuales que permite organizar tareas y colaborar en equipo de manera intuitiva. Sus principales funciones incluyen la creación de listas y tarjetas para gestionar el flujo de trabajo, la integración con otras aplicaciones a través de Power-Ups, y la capacidad de personalizar tableros para adaptarse a diferentes metodologías como Kanban. Sin embargo, algunos desafíos pueden incluir limitaciones en la gestión de proyectos complejos y la dependencia de integraciones adicionales para funcionalidades avanzadas (Atlassian, 2024e).

3.

GUÍA PARA LA GESTIÓN Y EL MANTENIMIENTO DEL **CICLO DE VIDA DE MODELOS DE PROYECTOS DE CIENCIA DATOS**

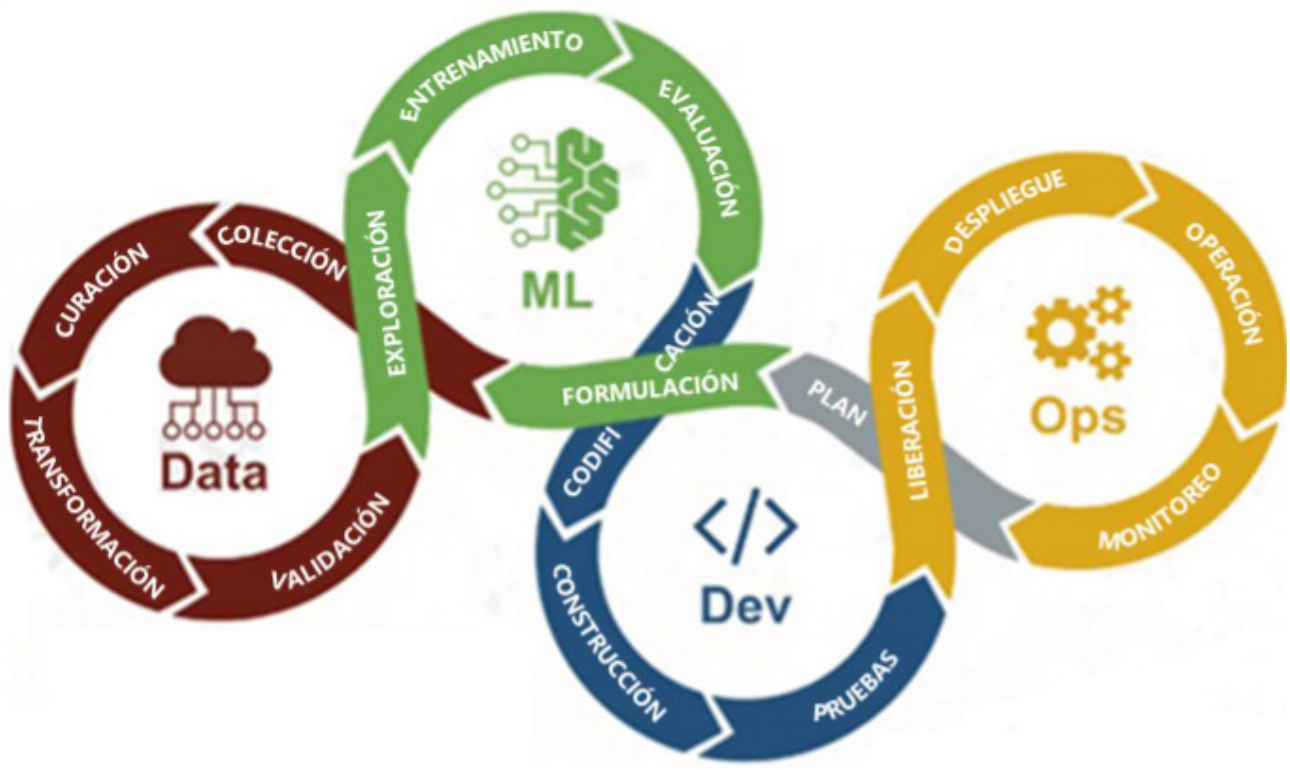
Considerando que, en este punto del documento, el lector ya posee un entendimiento profundo de los aspectos teóricos y conceptuales, así como de las herramientas de la infraestructura necesarias para gestionar las etapas del ciclo de vida de modelos en proyectos de ciencia de datos, se expone con un enfoque detallado y estructurado como mecanismo para la estandarización de proyectos de ciencia de datos con las mejores prácticas de documentación y codificación actuales de la industria. Se adhiere a los estándares de metodologías ágiles reconocidos a nivel internacional para la gestión de proyectos de ciencia de datos combinando las etapas del ciclo de vida de un modelo de ciencia de datos según MLOps (Machine Learning for Developers, 2022) con la estrategia de TDSP (Microsoft, 2016) para definir objetivos, tareas y artefactos en cada etapa.

La guía se estructura así: primero se explican detalladamente las etapas del ciclo de vida de un modelo en un proyecto de ciencia de datos, sus componentes clave y el orden de ejecución. Luego se presenta la estructura propuesta para un repositorio de este tipo de proyectos, acompañada de una explicación detallada de sus componentes y las mejores prácticas para la gestión de repositorios. Finalmente, se exponen plantillas de los artefactos clave que forman parte de la documentación de un proyecto de ciencia de datos. Se espera que al finalizar de leer esta guía, el lector cuente con los conocimientos necesarios para documentar sus proyectos de ciencia de datos de manera estandarizada y eficiente, utilizando las mejores prácticas de la industria.

3.1. Ciclo de vida de un modelo en un proyecto de ciencia de datos

Actualmente, un proyecto de ciencia de datos se origina a partir de la necesidad de desarrollar productos o funciones de productos impulsados por técnicas avanzadas de analítica de datos. Ya no se habla solo de un modelo individual, sino de una solución integral que preste atención meticulosa al diseño y funcionamiento óptimo de los modelos de ciencia de datos dentro del producto global. Por lo tanto, el desarrollo de modelos no puede realizarse de manera aislada debido a laas implicaciones críticas que puede tener en el producto y el negocio. Estas consideraciones se enmarcan en el ciclo propuesto por (Machine Learning for Developers, 2022) en la metodología MLOps compuesta por dieciséis etapas que se pueden visualizar de manera gráfica en la figura 2.

Figura 2. Etapas del ciclo de vida de un modelo con MLOps



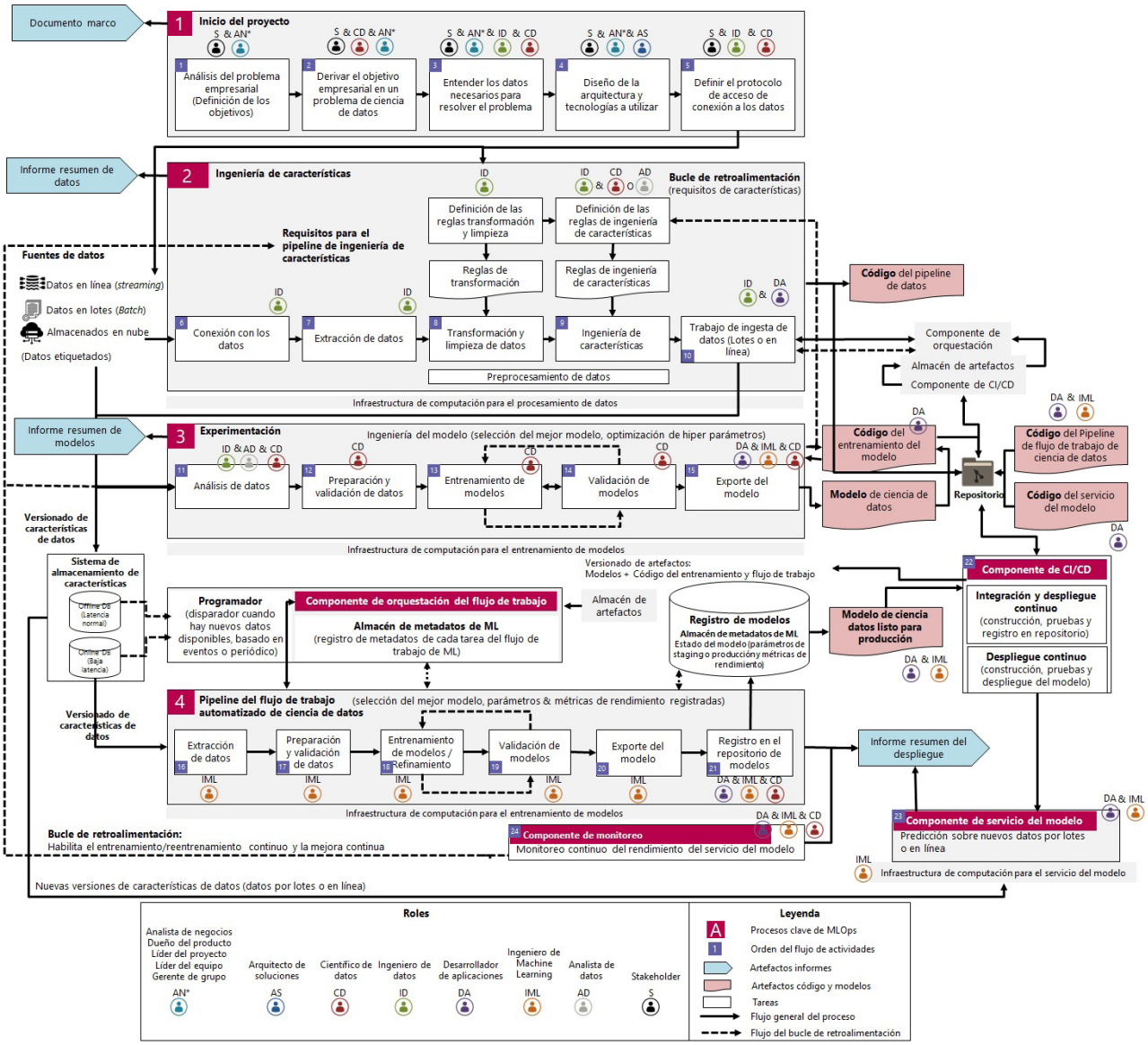
Fuente: (Machine Learning for Developers, 2022). Recuperado y traducido al español de https://www.ml4devs.com/articles/mlops-machine-learning-life-cycle/.

La presente guía simplifica la implementación de MLOps utilizando la propuesta de (Kreuzberger et al., 2023) de encapsular en cuatro procesos clave de forma iterativa el flujo completo de extremo a extremo. Parte desde el inicio del proyecto de ciencia de datos (1), sigue con el pipeline de ingeniería de características (2), la experimentación (3), hasta el pipeline del flujo de trabajo automatizado de ciencia de datos (4).

En la figura 3 se presenta la arquitectura conceptual con un flujo de trabajo sugerido para el desarrollo de un proyecto de ciencia de datos siguiendo MLOps. En esta, se visualizan los procesos clave en bloques horizontales numerados de for-

ma ordenada y consecutiva del 1 al 4. Dentro de cada bloque, se presentan las tareas en rectángulos blancos siguiendo un orden guiado por el flujo de las fechas. Los polígonos de flechas pentagonales azules representan los artefactos de informes, mientras que los rosas indican los de códigos y modelos. Los roles están representados por colores con un ícono humano. Las flechas con líneas de guiones representan el flujo de bucle o ciclo de retroalimentación. Los cilindros representan almacenamientos de características de los datos, modelos y sus metadatos. Los rectángulos magenta representan los componentes externos clave de orquestación, CI/CD, monitoreo y servicio del modelo. Todo lo anterior diseñado para resolver tareas específicas en el marco de asegurar el funcionamiento ininterrumpido de una solución integral enmarcada en un proyecto de ciencia de datos.

Figura 3. Arquitectura y flujo de trabajo de Machine Learning Operations de extremo a extremo con componentes funcionales, tareas, artefactos y roles



Fuente: DANE. Recuperado, adaptado y traducido al español de (Kreuzberger, D., Kuhl, N., & Hirschl, S., 2023); Machine Learning Operations (MLOps): Overview, Definition, and Architecture.

En continuación, para cada proceso se proponen los siguientes componentes clave, definidos como los elementos esenciales que permiten la ejecución correcta de cada proceso:

Tabla 3. Componentes claves de los procesos de Machine Learning Operations

Tareas, roles y responsabilidades	Las **tareas** corresponden a las actividades específicas necesarias que permiten el cumplimiento de los objetivos de los procesos clave. Los **roles** son el papel que desempeña cada miembro del equipo de trabajo. Las **responsabilidades** son las funciones asignadas a cada rol.
Artefactos	Los **artefactos** obedecen al conjunto de entregables resultantes de cada proceso clave.
Infraestructura, herramientas y utilidades	La **Infraestructura, herramientas** y **utilidades** comprenden el conjunto de componentes tecnológicos de hardware, software, plataformas, aplicaciones, marcos de trabajo y scripts sugeridos para el desarrollo de cada proceso clave.

Basado en la arquitectura y el flujo de trabajo previamente presentados, a continuación, se detalla de manera ordenada: 1) la descripción del proceso clave, 2) el listado de etapas de MLOps que lo componen, 3) las actividades que conforman cada tarea, los roles y responsabilidades, 4) los artefactos, y 5) la infraestructura, herramientas y utilidades recomendadas para la implementación y el desarrollo de cada uno de los procesos claves.

3.1.1. Inicio del proyecto de ciencia de datos

El inicio del proyecto es el proceso clave donde a partir de la evaluación del problema empresarial, se definen de forma precisa los objetivos del proyecto de ciencia de datos. Además, se planifica la infraestructura, se seleccionan las tecnologías adecuadas y se traducen los objetivos del negocio en problemas específicos de ciencia de datos, delimitando el alcance y estableciendo los resultados esperados. Este proceso clave desarrolla las etapas de planeación y formulación de MLOps en cinco tareas.

Tareas, roles y responsabilidades

A continuación, se presenta de forma ordenada la descripción de las tareas, actividades, roles y responsabilidades para su ejecución. Para efectos de entendimiento se recuerda que la leyenda de roles es:

1

Análisis del problema empresarial

Roles que intervienen

Actividades: evaluación detallada para identificar el problema y definir los objetivo y métricas del negocio. Propuesta de la hoja de ruta y características del producto para resolver el problema.

Desarrollo: el AN* establece mesas técnicas para el levantamiento de necesidades y requerimientos dictaminados por el S[23]. El AN* realiza preguntas claves como: ¿Cuál es el problema empresarial que pretende resolver?, ¿Cuál es el objetivo principal del proyecto desde la perspectiva empresarial?, ¿Quiénes son los principales stakeholders y usuarios finales del proyecto, y cuáles son sus necesidades y expectativas?, ¿Cuál es la justificación para optar por una solución de ciencia de datos? ¿Cuáles son los criterios y métricas que indicarán si la solución de ciencia de datos ha logrado los resultados esperados?, ¿Qué impacto se espera que tenga la solución de ciencia de datos en el negocio, y cómo se integrará en los procesos existentes? ¿Qué fechas límite se han establecido para la finalización del proyecto y qué debe incluirse en cada entregable para cumplir con las expectativas del cliente? Conforme a las respuestas del S, el AN* identifica el personal basado en roles que conformara su equipo de trabajo necesario para la ejecución del proyecto.

[23] Para efectos de esta guía se entenderá Stakeholder (S) como la persona o grupo que tiene interés en el proyecto y proporciona requisitos estratégicos. En ese sentido se debe entender como el usuario inicial solicitante del proyecto.

Actividades: traducción de los objetivos del negocio en problemas específicos de ciencia de datos. Investigación y evaluación del estado actual de los modelos dentro de la organización o de los avances en la industria, alineados con las necesidades estratégicas y los objetivos del negocio. Alineación de las métricas de rendimiento del modelo de ciencia de datos con las de negocio.

Desarrollo: el AN* en conjunto con el CD establecen el problema e hipótesis a solucionar a partir de modelos de ciencia de datos y se las presentan al S. Además, establecen los umbrales o criterios de éxito para las métricas del modelo de ciencia de datos en relación con las de negocio.

2

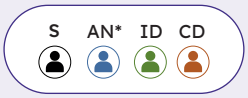
Derivar el objetivo empresarial en un problema de ciencia de datos

Roles que intervienen

3

Entender los datos necesarios para resolver el problema

Roles que intervienen



Actividades: identificación y evaluación de los datos requeridos para abordar el problema: Disponibilidad, accesibilidad y categorías de estos.

Desarrollo: el S le indica al ID y CD la disponibilidad de los datos o si se requiere de la captura de estos. El ID y CD realizan preguntas clave al S como: ¿Qué tipos de datos están disponibles para abordar el problema empresarial?, ¿Dónde se almacenan los datos y qué sistemas o bases de datos se utilizan para su gestión?, ¿Qué tamaño tienen los conjuntos de datos y en qué formatos están almacenados?, ¿Qué nivel de confidencialidad o sensibilidad tienen los datos, se requieren procesos de cifrado?, ¿Cómo se recopilan los datos y con qué frecuencia se actualizan o enriquecen?, ¿Cuál es el nivel de calidad de los datos disponibles y qué problemas han identificado?. Conforme a las respuestas del S, el ID y CD le comunican al AN* si el proyecto es técnicamente viable y las limitaciones que puede presentar.

Actividades: planificación de la infraestructura y selección de tecnologías adecuadas para el proyecto.

Desarrollo: a partir de los hallazgos de las tareas anteriores, el AS diseña la arquitectura de solución y la infraestructura de desarrollo de pruebas, valida la disponibilidad de los recursos tanto en el ambiente de desarrollo como de despliegue, y se la presenta al S. Además, el AN* define el presupuesto que abarca el coste del talento humano según los roles requeridos, así como la infraestructura y los recursos tecnológicos necesarios para el desarrollo del proyecto. De acuerdo con el presupuesto establecido, el AN* determina si el proyecto es viable desde el punto de vista financiero o si existen opciones de delimitación que permitan ajustar los costos para alcanzar la viabilidad.

4

Diseño de la arquitectura y tecnologías a utilizar

Roles que intervienen

5

Definir el protocolo de acceso de conexión a los datos

Roles que intervienen

Actividades: identificar la localización de los datos y definir el protocolo de acceso.

Desarrollo: el S le indica al ID y CD la ubicación y tipo de datos para que estos establezcan el protocolo de acceso de conexión a estos en su estado original.

Artefactos

El artefacto o entregable de este proceso clave es el Anexo A. Documento Marco (project_charter.md). Este define el alcance, los objetivos y los requisitos del proyecto de ciencia de datos. Además, establece la visión del proyecto, los criterios de éxito, los roles y responsabilidades, así como el plan de trabajo detallado, asegurando una alineación efectiva entre los objetivos empresariales y los esfuerzos técnicos. Igualmente, incluye la arquitectura de solución y el presupuesto del proyecto. También incluye el resumen ejecutivo, los principales resultados, las lecciones aprendidas, el impacto, las conclusiones y los agradecimientos del proyecto.

Infraestructura, herramientas y utilidades

Se recomienda utilizar herramientas de gestión de proyectos como Jira, Grafana, Notion, Confluence o Azure DevOps. Esta última es sugerida actualmente en el DANE, ya que se integra de manera eficiente en el ecosistema de Microsoft, incluyendo Office 365 y Azure Cloud, optimizando así los flujos de trabajo.

3.1.2. Ingeniería de características

La canalización de ingeniería de características es un proceso clave cuyo objetivo es automatizar los pasos para transformar los datos originales en nuevas características útiles. Este proceso se basa en requisitos definidos para generar datos preprocesados y limpios. Las fórmulas y transformaciones necesarias son introducidas por el ingeniero de datos en colaboración con el científico de datos, asegurando así la relevancia y efectividad de las nuevas características para mejorar los modelos de ciencia de datos. Es importante señalar que este proceso es iterativo, la aplicación de un conjunto de características puede generar nuevas ideas, que se implementan en otro conjunto de características, y así indefinidamente. Esto se muestra claramente en el esquema como un circuito de retroalimentación. Este proceso clave desarrolla las etapas de colección, curación y transformación de MLOps en cinco tareas.

Tareas, roles y responsabilidades

A continuación, se presenta de forma ordenada la descripción de las tareas, actividades, roles y responsabilidades para su ejecución. Para efectos de entendimiento se recuerda que la leyenda de roles es:

6
Conexión con los datos

Roles que intervienen

ID

Actividades: establecer acceso y conexión a las fuentes de datos en su estado original que pueden ser de tipo : en vivo (on-streaming), por lotes (batch) o alojados en algún Sistema de almacenamiento de datos. En el caso donde la fuente de datos sea nueva, se deberá crear la fuente correspondiente y configurar la conexión necesaria para su integración.

Desarrollo: el ID identifica las fuentes de datos y los requisitos para establecer la conexión a estos como proceso inicial del pipeline.

7
Extracción de datos

Roles que intervienen

ID

Actividades: obtener los datos de las conexiones previamente definidas para su posterior análisis. Para aquellos proyectos que requieran enriquecer sus datos a partir de fuentes secundarias se consideran estrategias como Web scraping a servicios web, redes sociales, aplicaciones móviles, entre otros.

Desarrollo: el ID establece el proceso para extraer los datos de las fuentes identificadas y con conexión validad en el proceso del pipeline anterior.

8
Transformación y limpieza de datos

Roles que intervienen

ID

Actividades: procesamiento de datos para corregir errores, eliminar inconsistencias y formateo adecuado para garantizar la integridad de estos basado en un análisis preliminar exploratorio. Incluye la normalización, agregación, y demás reglas establecidas para la limpieza de datos.

Desarrollo: el ID define la reglas de transformación y limpieza de datos para llevar los datos extraídos en el proceso anterior a un formato utilizable.

Actividades: creación de nuevas variables a partir de datos existentes, optimizando así la representación de los datos para mejorar el rendimiento del modelo. Incluye la selección de atributos relevantes que pueden aumentar la precisión predictiva y la eficiencia del modelo. Los roles que intervienen en esta tarea son principalmente: el ingeniero de datos y científico de datos.

Desarrollo: el CD y AD en común acuerdo definen las reglas de ingeniería de características, que son comunicadas al ID para que este las implemente. Las reglas inicialmente definidas se deben ajustar iterativamente por el CD, basándose en la retroalimentación de la etapa de ingeniería del modelo experimental y del componente de monitoreo que observa el desempeño del modelo.

9

Ingeniería de características

Roles que intervienen

ID CD AD

10

Trabajo de ingesta de datos

Roles que intervienen

ID DA

Actividades: recopilación y carga de datos ya sea por lotes (batch) o en línea (on-streaming) en el sistema de almacenamiento de características, este puede ser una base de datos en línea o fuera de línea.

Desarrollo: el ID enlaza este última etapa al pipeline el cual debe encontrarse totalmente automatizado y el cual genera un código como artefacto del segundo proceso clave. Y se comunica con el DA para integrar este pipeline en una orquestación de CI/CD.

Artefactos

Los artefactos o entregables de este proceso clave son el Anexo B. Informe resumen de datos (data_summary.md) y el código y la documentación del Pipeline de datos. El informe integra la definición detallada de los datos, su diccionario y un resumen exhaustivo. Proporciona una descripción completa de las características de los datos, incluyendo su estructura, tipos, y relaciones. Además, ofrece un diccionario de datos que define cada variable y su significado, junto con un resumen que destaca las principales estadísticas y hallazgos relevantes, facilitando una comprensión clara y precisa para todo el equipo de ciencia de datos.

Infraestructura, herramientas y utilidades

Se recomienda utilizar herramientas de gestión y almacenamiento de datos dependiendo del tipo de datos. Para datos estructurados se sugiere utilizar MySQL, PostgreSQL, Microsoft SQL Server o Oracle Database. Esta última sugerida actualmente en el DANE, debido a su eficiencia en el manejo de grandes volúmenes de datos, su robustez en transacciones complejas, y su capacidad para integrarse con otras herramientas empresariales. Para no estructurados se propone el uso de Apache Cassandra, Hbase, Redis o MongoDB. Esta última recomendada actualmente en la Entidad, debido a su flexibilidad para manejar datos semiestructurados, su escalabilidad horizontal y su capacidad para realizar consultas rápidas y eficientes. Para almacenamiento distribuidos se sugieren Hadoop HDFS, Amazon S3, Google Cloud Storage o Azure Storage Account. Esta última sugerida actualmente en la Entidad, ya que se integra de manera eficiente en el ecosistema de Microsoft, incluyendo Azure Cloud.

De igual forma, se recomienda utilizar herramientas de procesamiento de datos específicas según el tipo de tarea requerida. Para procesamiento en lote se sugiere usar Apache Spark, Google Dataflow, Amazon EMR o Azure Batch. Esta última sugerida actualmente en el DANE, ya que se integra de manera eficiente en el ecosistema de Microsoft, incluyendo Azure Cloud. Para procesamiento en tiempo real se propone el uso de Apache Kafka, Apache Flink, Amazon Kinesis o Azure Stream Analytics. Esta última sugerida actualmente en la entidad, ya que se integra de manera eficiente en el ecosistema de Microsoft, incluyendo Azure Cloud.

Por otro lado, se recomienda utilizar herramientas de perfilamiento y limpieza de datos como Talend Data Quality, Trifacta, Great Expectations o OpenRefine. Esta última sugerida actualmente en el DANE, por su facilidad para la normalización de datos desestructurados, la detección y la corrección de inconsistencias y mejora significativamente la calidad de los datos antes de su análisis. Finalmente, se recomienda utilizar herramientas de versionado de datos como Quilt Data, Pachyderm o Data Version Control – DVC. Esta última sugerida actualmente en el DANE, por su gran facilidad de uso, por ser de uso abierto y porque permite la integración perfectamente con los flujos de trabajo existentes basados en GIT.

3.1.3. Experimentación

Este es un proceso clave cuyo objetivo es desarrollar un modelo de ciencia de datos funcional que pueda abordar las necesidades del usuario alineándose con los objetivos empresariales. Esto implica iterar sobre diferentes enfoques y técnicas para optimizar el rendimiento del modelo, asegurando su efectividad en resolver problemas específicos del negocio. Este proceso clave desarrolla las etapas de validación, exploración, entrenamiento y evaluación de MLOps en cuatro tareas.

Tareas, roles y responsabilidades

A continuación, se presenta de forma ordenada la descripción de las tareas, actividades, roles y responsabilidades para su ejecución. Para efectos de entendimiento se recuerda que la leyenda de roles es:

11 Análisis de datos

Roles que intervienen

ID

Actividades: realizar el Análisis Exploratorio de datos (EDA) para identificar patrones, tendencias y relaciones entre diversas características y la variable objetivo. Esto incluye estadísticas descriptivas, visualización de datos, distribución y correlación de variables.

Desarrollo: el AD y CD en apoyo del ID se conectan al sistema de almacenamiento de características para realizar el EDA.

12 Preparación y validación de datos

Roles que intervienen

CD

Actividades: preprocesamiento para asegurar un conjunto de datos con alta calidad antes del modelado. Incluye la detección de anomalías en los datos mediante la aplicación de reglas específicas alineadas con los requisitos del negocio como: Control sobre posibles desviaciones en la distribución estadística de los datos en el tiempo, cambios inesperados en correlaciones o patrones entre variables, valores faltantes, formatos inválidos y sesgo en los datos que puedan influir en los resultados de los modelos. También incluye la creación de conjuntos de datos divididos en entrenamiento, prueba y validación.

Desarrollo: el CD elabora las reglas de preparación y validación de datos, y genera el código correspondiente para implementarlas.

13
Entrenamiento de modelos

Roles que intervienen

ID

Actividades: aplicación de algoritmos de ciencia de datos utilizando los datos preprocesados en las etapas previas.

Nota: *es recomendado utilizar el principio de parsimonia (navaja de OCCAM), comenzando con modelos simples y progresando hacia modelos más complejos. Se realiza el entrenamiento, ajuste de hiper parámetros y selección de mejores modelos.*

Desarrollo: el CD experimenta con diferentes algoritmos sobre los datos de entrenamiento. Se puede apoyar en el ID para optimizar el código de entrenamiento de modelo bien diseñado.

14
Validación de modelos

Roles que intervienen

CD

Actividades: realizar la ingeniería de modelos para identificar el algoritmo y los hiper parámetros de mejor rendimiento para el modelo.

Desarrollo: el CD experimenta diferentes parámetros del modelo de forma interactiva en el entrenamiento del modelo hasta alcanzar las métricas de rendimiento previamente definidas como óptimas, tanto a nivel de eficacia del modelo como de eficiencia computacional, asegurando así resultados satisfactorios. La tarea de entrenamiento y validación del modelo se pueden repetir iterativamente.

15 Exporte del modelo

Roles que intervienen

Actividades: guardar y preparar en el repositorio corporativo el modelo entrenado que cumpla el rendimiento y las métricas de negocio establecidas en el primer proceso clave para su implementación en entornos de producción. Así como el código, los parámetros seleccionados y la documentación del modelo.

Desarrollo: el CD exporta el modelo y envía el código al repositorio.

Nota: *el IML define el código para la canalización del flujo de trabajo automatizado de ciencia de datos y lo envía al repositorio. Para que cada vez que el CD envíe un nuevo modelo o el IML envíe un nuevo código de canalización de flujo de trabajo al repositorio, el componente CI/CD detecte el código actualizado y active automáticamente la canalización de CI/CD que lleva a cabo los pasos de compilación, prueba y entrega. El paso de compilación crea artefactos que contienen el modelo y las tareas de la canalización del flujo de trabajo. El paso de prueba valida los criterios de calidad seleccionados para la aceptación de la solución, el modelo y el código de canalización del flujo de trabajo. Mientras que el paso de entrega envía los artefactos versionados, al almacén de artefactos.*

Artefactos

Los artefactos o entregables de este proceso clave son el Anexo C. Informe resumen de modelos (model_summary.md), los modelos e hiper parámetros, los datos procesados, el código y documentación del entrenamiento del modelo. El informe documenta el punto de partida con el modelo base y describe el proceso y resultados obtenidos con el modelo final tras la experimentación. Proporciona una evaluación detallada del desempeño de los modelos, incluyendo métricas clave, comparaciones y mejoras logradas, ofreciendo una visión clara del desarrollo, la eficacia de los modelos implementados y su selección final.

Infraestructura, herramientas y utilidades

Se recomienda utilizar herramientas de desarrollo y entrenamiento de modelos dependiendo del nivel de madurez de recursos con los que cuente la Entidad. Si se trata de experimentación en entornos de la nube las opciones son Azure Databricks, Amazon SageMaker, Vertex AI o Google Colab. Si se realiza en un entorno local se puede utilizar Jupyter Notebook, Visual Studio Code, Pycharm o RStudio. Es importante destacar que la experimentación con modelos de ciencia de datos requiere significativos recursos computacionales, incluidos GPU. Por esta razón, se recomienda utilizar un entorno en la nube para este proceso clave, ya que ofrece la escalabilidad y la potencia necesarias para manejar las intensivas cargas de trabajo de manera eficiente. Se sugiere Azure Databricks actualmente en el DANE, ya que se integra de manera eficiente en el ecosistema de Microsoft, incluyendo Azure Cloud.

Por otra parte, se recomienda utilizar herramientas de visualización de datos dependiendo del nivel de madurez de recursos con los que cuente la Entidad y la complejidad del proyecto. Si se cuenta con recursos para plataformas se sugiere el uso de Tableau, Looker Studio, Amazon QuickSight, Oracle Analytics o PowerBI. Esta última sugerida actualmente en el DANE, ya que se integra de manera eficiente en el ecosistema de Microsoft, incluyendo Azure Cloud. Si por el contrario se opta por un desarrollo se sugiere el uso de bibliotecas de Python y R como Plotly Dash, Bokeh, Shiny o Flexdashboard.

Finalmente, se recomienda utilizar herramientas de versionado de código como Gitea, GitHub, Bitbucket, Apache Subversion – SVN o Gitlab. Esta última sugerida actualmente en la Entidad, ya que se integra de manera eficiente en el ecosistema de Microsoft, incluyendo Azure Cloud y por qué cuenta con un componente de CI/CD que optimiza los flujos de trabajo y facilita la automatización de pruebas, despliegues y otros procesos críticos.

3.1.4. Pipeline del flujo de trabajo automatizado de ciencia de datos y servicio y monitoreo del modelo

El flujo de trabajo automatizado de ciencia datos es un proceso clave cuyo objetivo es configurar una canalización o pipeline que, basándose en métricas y sin intervención humana, inicie automáticamente procesos de reentrenamiento o experimentación con diferentes modelos, eligiendo eventualmente el mejor que reemplace al actual en producción. Este proceso es el más robusto de todo ya que involucra varios componentes externos clave.

- El componente del orquestador de flujo de trabajo es responsable de ejecutar la canalización a partir de en un evento o un horario específico.

- El sistema de almacenamiento de características proporciona los datos necesarios sobre las características requeridas para el modelo, facilitando el acceso y la consistencia de los datos.

- El registro de modelos y el almacén de metadatos de ML almacena los modelos y sus métricas obtenidas tras la finalización del proceso, permitiendo el seguimiento y la gestión eficiente de los modelos y sus versiones.

Este proceso clave desarrolla las etapas de codificación, construcción, pruebas, liberación, despliegue, operación y monitoreo de MLOps en nueve tareas.

Tareas, roles y responsabilidades

A continuación, se presenta de forma ordenada la descripción de las tareas, actividades, roles y responsabilidades para su ejecución. Para este proceso clave, el Ingeniero de Machine Learning se encarga de la gestión del flujo de trabajo automatizado de ciencia de datos, así como de la infraestructura de entrenamiento del modelo subyacente en forma de recursos de hardware y marcos de trabajo que respaldan el procesamiento. El componente de orquestación coordina y organiza las tareas del flujo de trabajo automatizado recopilando los metadatos para cada tarea en forma de registros, tiempo de finalización, etc. Para cada tarea, los artefactos necesarios son extraídos del almacén de artefactos. Para efectos de entendimiento se recuerda que la leyenda de roles es:

16

Extracción de datos

Roles que intervienen

IML

Actividades: extracción automatizada de las características versionadas en los sistemas de almacenamiento de características que pueden extraerse de la base de datos en línea o fuera de línea (o de cualquier tipo de almacén de datos).

Desarrollo: el programador (trigger) detecta cuando hay nuevos datos disponibles basado en algún evento o programación periódica activando el pipeline del flujo de trabajo automatizado de ciencia de datos.

Actividades: preparación y validación automatizada de datos basada en las reglas definidas en la etapa de experimentación.

Desarrollo: una vez finaliza la etapa 16. Extracción de datos automatizada esta tarea se ejecuta de forma automática.

17

Preparación y validación de datos

Roles que intervienen

IML

18

Entrenamiento de modelos / refinamiento

Roles que intervienen

IML

Actividades: entrenamiento automatizado del modelo final sobre nuevos datos no observados. El algoritmo y los hiperparámetros ya están predefinidos en función de la configuración de la etapa de experimentación anterior. El modelo es reentrando y refinado.

Desarrollo: una vez finaliza la etapa 17. Preparación y validación automatizada de datos esta tarea se ejecuta de forma automática.

Actividades: realizar la ingeniería de modelos de forma automatizada para identificar el algoritmo y los hiperparámetros de mejor rendimiento para el modelo.

Desarrollo: una vez finaliza la etapa 18. Entrenamiento y refinamiento automatizado del modelo de datos esta tarea se ejecuta de forma automática. La tarea de entrenamiento de modelo automatizado y de validación de modelo automatizado se repiten iterativamente hasta alcanzar las métricas de rendimiento previamente definidas como óptimas, tanto a nivel de eficacia del modelo como de eficiencia computacional, asegurando así resultados satisfactorios.

19

Validación de modelos

Roles que intervienen

IML

20
Exporte del modelo

Roles que intervienen

IML

Actividades: guardar y preparar automáticamente en el repositorio corporativo el modelo entrenado que cumpla el rendimiento y métricas de negocio establecidas en el primer proceso clave para su implementación en entornos de producción. Así como el código y los parámetros seleccionados.

Desarrollo: una vez finaliza la etapa 19. Validación automatizada del modelo de datos esta tarea se ejecuta de forma automática.

21
Registro en el repositorio de modelos

Roles que intervienen

DA IML CD

Actividades: registro del modelo y sus artefactos en el almacén de metadatos de ML. Esto también se denomina registro del linaje de modelos el cual combina el versionado de datos y de código para cada modelo registrado, así como del estado del modelo: en preparación (staging) o listo para producción (production-ready).

Desarrollo: una vez finalizada la etapa 20. Exporte del modelo, se debe ejecutar esta tarea de forma automática.

22
Paso a producción

Roles que intervienen

DA IML

Actividades: preparar la canalización de integración y el despliegue continuo que extraiga, construya, pruebe y despliegue el modelo y el código de servicio del modelo en producción. Esto debe incluir una estrategia de despliegue acorde con la infraestructura definida (por ejemplo, despliegue total o basado en canarios[24]).

Desarrollo: una vez finaliza la etapa 21. Registro en el repositorio, el modelo es entregado de forma automática al IML para su implementación en la infraestructura computacional habilitada junto con el DA generalmente en aplicaciones configuradas en un contenedor, bibliotecas, APKs, Servicios Web, APIs, entre otras. Se activa el proceso de integración y despliegue continuo del componente de CI/CD, en el cual la canalización de integración y despliegue continuo realiza la construcción y la prueba del modelo y el código de servicio, y lo despliega para su uso en producción.

[24] El despliegue basado en canarios es un enfoque de implementación gradual de software en producción que implica lanzar una nueva versión de software a un pequeño grupo de usuarios antes de hacerlo para toda la base de usuarios (Amazon Web

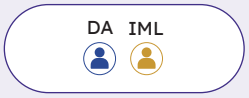
Actividades: gestionar el despliegue asegurando un rendimiento óptimo y permitiendo ajustes o reversiones según sea necesario. Para ello se debe realizar de forma continua y transversal la evaluación de: integración, rendimiento y robustez de los modelo, así como a la implementación de pruebas de seguridad (vulnerabilidades y privacidad).

Desarrollo: una vez finaliza la etapa 22. Paso a producción, se activa el componente de servicio, el modelo está en la capacidad de generar predicciones a través de un llamado generalmente de un API REST ya sea con datos nuevos o no observados provenientes del sistema de almacenamiento de características que pueden ser en tiempo real (on-streaming) o por lotes (batches).

23

Operacionalización del modelo

Roles que intervienen



24

Monitoreo del modelo

Roles que intervienen

Actividades: supervisar de manera continua el rendimiento de la solución, incluyendo la salud del sistema, la detección de errores, las latencias, las métricas del modelo, garantizando así el funcionamiento óptimo y la calidad de la solución desplegada.

Desarrollo: el componente de monitoreo (previamente configurado por el IML, DA y CD) se encuentra activo continuamente, es el que se encarga de observar de forma continua el rendimiento del servicio del modelo y la infraestructura en tiempo real. Si este detecta una degradación en las métricas de rendimiento, la información se transmite a través del bucle de retroalimentación, habilitando el reentrenamiento y mejora del modelo, hace que la información se transfiera a varios puntos receptores, como la etapa experimental, el pipeline de ingeniería de datos y el programador. La retroalimentación a la etapa experimental es gestionada por el CD para mejoras adicionales del modelo. La retroalimentación a la zona de ingeniería de datos permite el ajuste de las características preparadas para el sistema de almacenamiento de características.

Artefactos

Los artefactos o entregables de este proceso clave son el Anexo D. Informe resumen de despliegue (deployment_summary.md), los modelos e hiper parámetros, los datos procesados, el código y documentación del entrenamiento del modelo. El informe documenta detalladamente el proceso de implementación del modelo de ciencia de datos en producción. Incluye la descripción de las etapas de despliegue, las configuraciones utilizadas, los resultados de las pruebas realizadas y cualquier incidencia encontrada. Este informe proporciona una visión clara y concisa del estado del modelo desplegado, facilitando el seguimiento y la gestión del rendimiento en producción.

Infraestructura, herramientas y utilidades

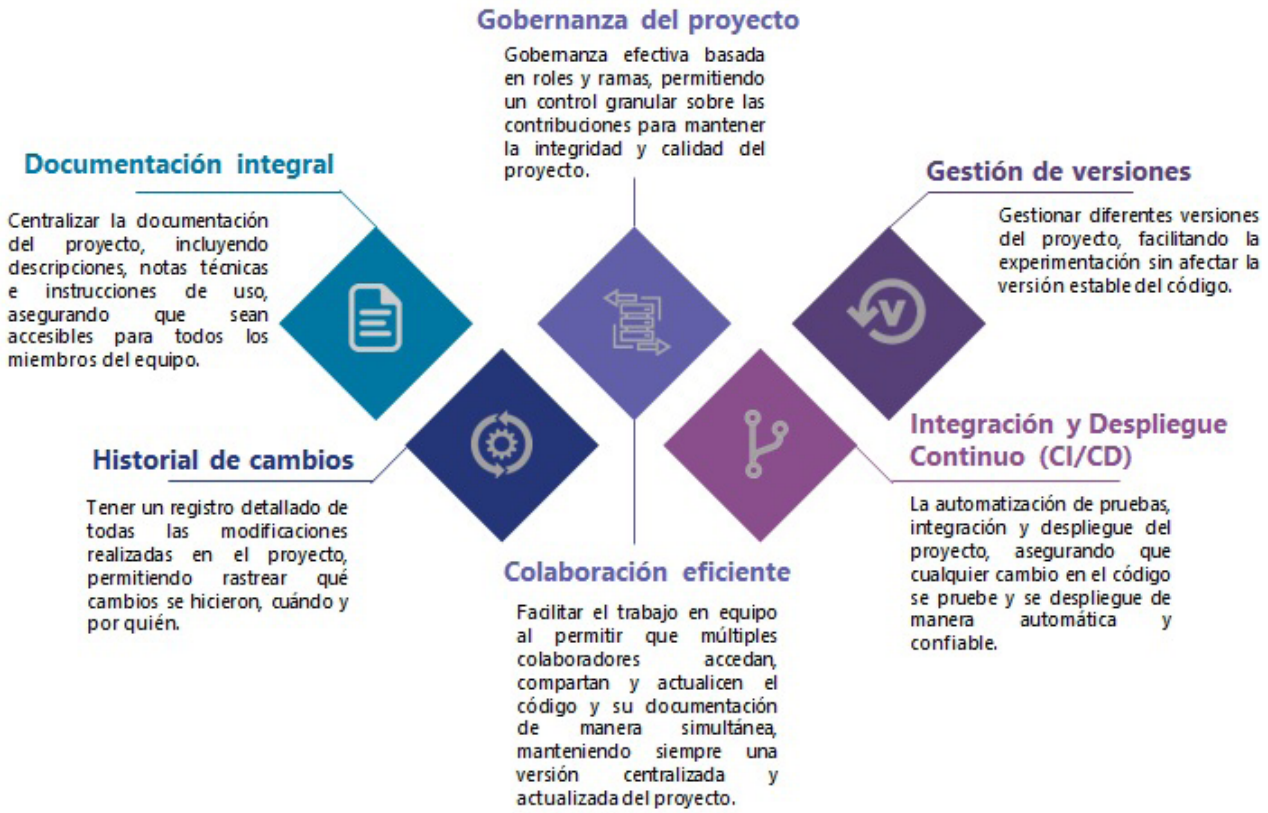
Se recomienda utilizar herramientas de MLOps y CI/CD dependiendo del nivel de madurez de recursos con los que cuente la Entidad. Para integración continua se recomienda el uso de Jenkins, Travis o CircleCI. Mientras que para despliegue continuo se sugiere utilizar Kubernetes, Terraform, Ansible o Docker. Por otra parte, para operación y mantenimiento del modelo se sugiere el uso de MLflow, Kubeflow, Azure DevOps, Seldon o Tecton. Se sugiere el uso de MLflow en la entidad para despliegues simples debido a sus capacidades como herramienta open source, su facilidad de integración y su soporte para el seguimiento de experimentos, gestión de modelos y registro de artefactos. Por otro lado, se recomienda Azure DevOps si el desarrollo completo se ha realizado en el entorno de Azure Cloud, debido a su integración robusta con otros servicios de Azure y sus avanzadas capacidades de gestión y automatización de proyectos.

Por otra parte, se propone el uso de herramientas de orquestación de flujos de trabajo como Apache NiFi, Kubernetes, Luigi, Prefect, Dagster o Apache Airflow. Esta última sugerida actualmente en el DANE, ya que: cuenta con una interfaz gráfica intuitiva y una sintaxis basada en Python; es una herramienta madura y ampliamente adoptada en la industria; es open source, y ofrece amplias integraciones con múltiples servicios y herramientas, incluyendo bases de datos, servicios de cloud, y sistemas de procesamiento de datos.

3.2. Repositorio para un proyecto de ciencia de datos

Un repositorio de un proyecto de ciencia de datos es una estructura organizada de directorios, archivos y código, que cuentan con el historial de los cambios que han sufrido a lo largo del ciclo de vida del proyecto. Las principales funcionalidades del repositorio están descritas a continuación:

Figura 4. Funcionalidades de un repositorio para un proyecto de ciencia de datos



Fuente: DANE.

Para fines de esta guía, en la figura 5 se presenta una propuesta de estructura de repositorio de un proyecto que contempla los cuatro procesos clave que componen el ciclo de vida de un modelo de ciencia de datos. Este se compone por 12 directorios que contienen los archivos necesarios para gestionar un proyecto completo de ciencia de datos:

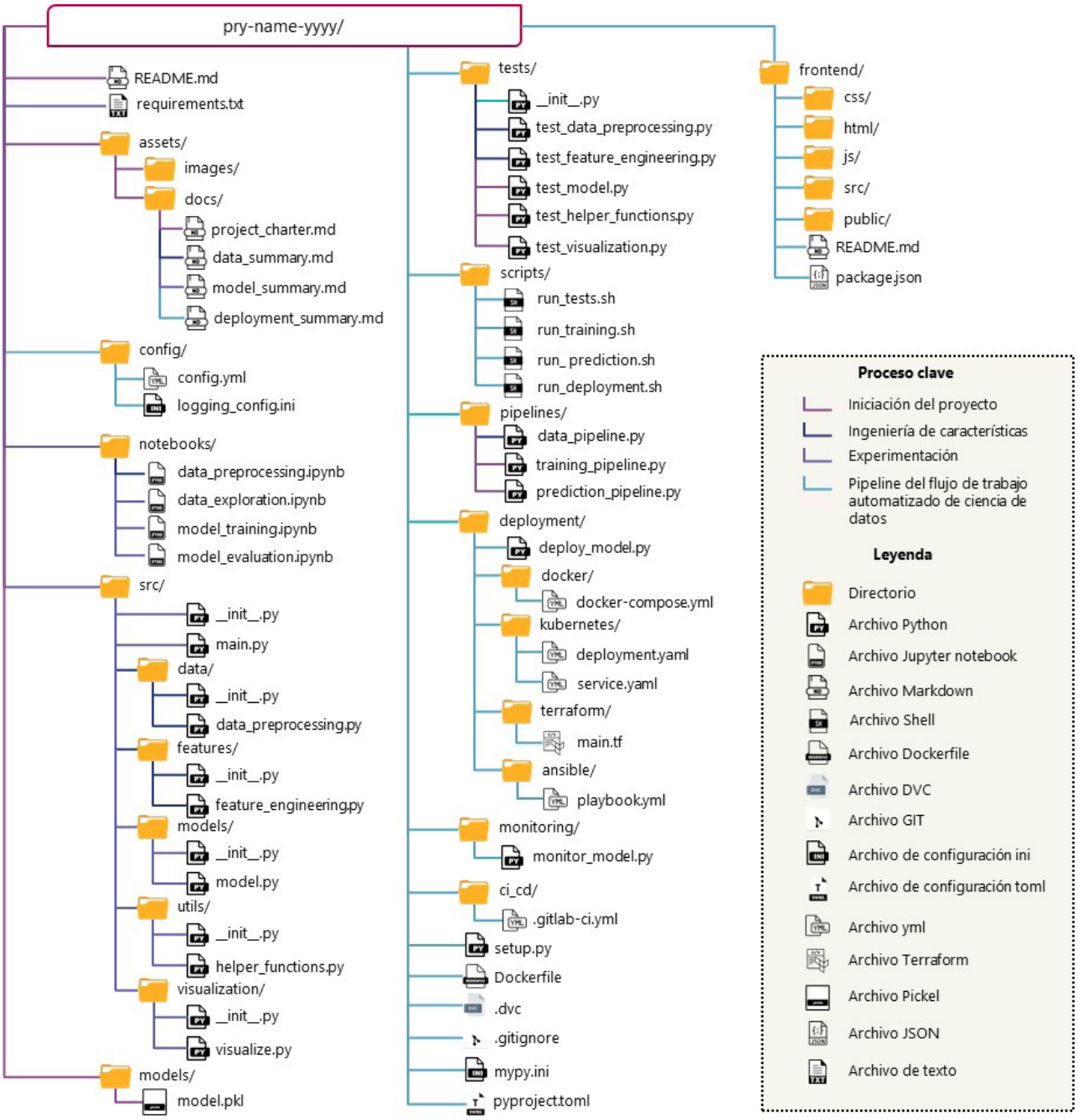
assets	Almacena archivos estáticos y recursos adicionales como imágenes y documentos.
config	Contiene los archivos de configuración del proyecto, incluyendo configuraciones de logs.
notebooks	Contiene los notebooks de Jupyter para exploración de datos, preprocesamiento, entrenamiento y evaluación de modelos
src	Contiene el código fuente del proyecto, incluyendo scripts para el preprocesamiento de datos ingeniería de características, definición y entrenamiento de modelos, visualización y funciones auxiliares.
tests	Contiene las pruebas unitarias y de integración para asegurar la funcionalidad y calidad del código, con scripts específicos para preprocesamiento de datos, ingeniería de características, modelos y funciones auxiliares.
scripts	Contiene scripts de automatización para ejecutar tareas específicas de validación de pruebas, entrenamiento del modelo, ejecución de predicciones y despliegue del modelo.
pipelines	Contiene las canalizaciones para el preprocesamiento de datos, entrenamiento del modelo y predicciones.
deployment	Contiene los Scripts y configuraciones para el despliegue del modelo ya sea en Docker, Kubernetes, Terraform o Ansible.
monitoring	Contiene los Scripts y herramientas para la monitorización del modelo en producción.
ci/cd	Contiene las configuraciones y scripts para integración continua (CI) y despliegue continuo (CD) con GitLab CI.
frontend	Contiene el Código del front-end del proyecto, con configuraciones de Webpack y dependencias.
models	Almacena los modelos entrenados y sus hiper parámetros de entrenamiento.

Fuera de estos directorios se encuentran archivos necesarios para la presentación y las instrucciones del proyecto:

README.md	Archivo de texto escrito en formato Markdown que proporciona una descripción detallada del proyecto. Es el primer archivo que los usuarios ven cuando visitan el repositorio y sirve como una guía completa para entender, instalar, usar y contribuir al proyecto.
requirements.txt	Archivo que contiene el listado de dependencias del proyecto.
setup.py	Archivo de configuración estándar para empaquetar y distribuir proyectos de Python. Define cómo se empaqueta el proyecto y sus dependencias, y se utiliza junto con herramientas como setuptools y pip.
Dockerfile	Script que contiene una serie de instrucciones para construir una imagen Docker. Cada instrucción en el Dockerfile crea una capa en la imagen y empaqueta todas estas capas en una imagen que se puede ejecutar en cualquier entorno que soporte Docker.
.dvc	Archivo de configuración del versionado de datos en Data Version Control.
.gitignore	Archivo que configura los archivos y carpetas a ignorar por git.
mypy.ini	Archivo para configurar MyPy, una herramienta de chequeo de tipos estáticos para Python. Permite definir reglas y configuraciones para cómo MyPy debe verificar el código del proyecto
pyproject.toml	Archivo de configuración estándar para proyectos de Python que define información sobre el proyecto y sus dependencias. Es utilizado por varias herramientas de construcción y empaquetado, como poetry para gestionar las dependencias y configuraciones del proyecto

En particular, esta estructura permite que un proyecto pueda ser dockerizado y desplegado ya sea en Kubernetes, Ansible o Terraform. La configuración incluye scripts y definiciones de pipelines para automatizar el flujo de trabajo desde la ingesta de datos hasta el despliegue y monitorización del modelo en producción. Con el soporte de Docker, Kubernetes, Ansible o Terraform, este proyecto puede ser fácilmente integrado en aplicaciones web o sistemas de información, proporcionando capacidades avanzadas de análisis y predicción de datos en un entorno de alta disponibilidad y resiliencia.

Figura 5. Estructura en árbol de carpetas y archivos de un repositorio para un proyecto de ciencia de datos



Fuente: DANE.

Se aclara, que esta estructura de repositorio está pensada para el desarrollo de un proyecto de ciencia de datos desde cero in-house, pero también es lo suficientemente general y flexible para adaptarse a diferentes entornos de desarrollo y despliegue según los requisitos específicos del proyecto y las herramientas y las plataformas que se utilicen para su implementación.

3.2.1. Mejores prácticas para la gestión de repositorios

Como fue mencionado previamente, un repositorio es una estructura de datos organizada que facilita el trabajo simultáneo y colaborativo en proyectos de ciencia de datos. Sin embargo, dado este sirve como entorno compartido, pueden surgir diversos desafíos en la gestión efectiva de las contribuciones realizadas por cada participante. A raíz de esto se presentan los siete principales practicas recomendadas por (GitHub Incorporated, 2024b; Gitlab incorporated, 2024b) para la gestión de repositorios.

a) Configurar el usuario Git, clonar el repositorio y trabajar de forma remota

El primer paso será establecer su identidad en Git mediante la configuración de su nombre de usuario y correo electrónico. Esto asegura que todos las confirmaciones *(commits)* que se realicen estén correctamente etiquetados con la información de su autoría. Para ello utilizar los comandos:

```
git config --global user.name <"Tu Nombre">
git config --global user.email <"tu.email@example.com">
```

Luego de esto, ya se puede clonar el repositorio y obtener una copia local del repositorio remoto. Esto le permitirá trabajar en su entorno local mientras se mantiene una sincronización con el repositorio central. Para ello se utiliza el comando:

```
git clone <url-del-repositorio>
```

Se debe verificar que el repositorio local esté correctamente vinculado al repositorio remoto usando:

```
git remote -v
```

b) Nombrar correctamente las ramas de acuerdo con su propósito

Nombrar las ramas de manera clara y consistente es fundamental para mantener un flujo de trabajo organizado y comprensible en proyectos colaborativos. Por ello se sugiere utilizar las siguientes convenciones de nombres, diferenciadas por el carácter "/" para organizar y categorizar las ramas de forma jerárquica:

Tabla 4. Convención para el nombramiento de ramas en un repositorio

master or main	Es la rama principal y refleja el **estado de producción** del proyecto. Contiene el código estable y listo para ser lanzado a producción. **No se deben realizar desarrollos directamente en esta rama**	
develop	Es la rama principal de **desarrollo** donde se integran todas las características y correcciones antes de ser preparadas para el lanzamiento. Es la base sobre la cual se desarrollan nuevas funcionalidades y se realizan pruebas antes de fusionar los cambios a **master** o **main.**	
release/[number.number.number]	Son las ramas creadas para preparar una versión específica del proyecto antes de su lanzamiento. Permiten realizar pruebas finales, correcciones menores y ajustes específicos para esa versión.	**Ejemplo:** release/0.0.1
feature/[descriptive-name]	Son las ramas dedicadas al desarrollo de nuevas características o funcionalidades. Cada rama debe estar asociada a una tarea o historia de usuario específica y debe tener un nombre descriptivo que indique la funcionalidad que se está desarrollando.	**Ejemplo:** feature/add-data-augmentation
hotfix/[descriptive-name]	Son las ramas utilizadas para realizar correcciones críticas y urgentes directamente en la rama **master** o **main.** Se emplean cuando es necesario solucionar un problema crítico que afecta a la producción y no puede esperar a la próxima liberación.	**Ejemplo:** hotfix/resolve-model-deployment-error
bugfix/[descriptive-name]	Son las ramas dedicadas a la corrección de errores encontrados durante el desarrollo. Se crean a partir de **develop** y se utilizan para abordar problemas específicos *(issues)* antes de que se fusionen en la rama **develop.**	Ejemplo: bugfix/resolve-validation-metric-issue

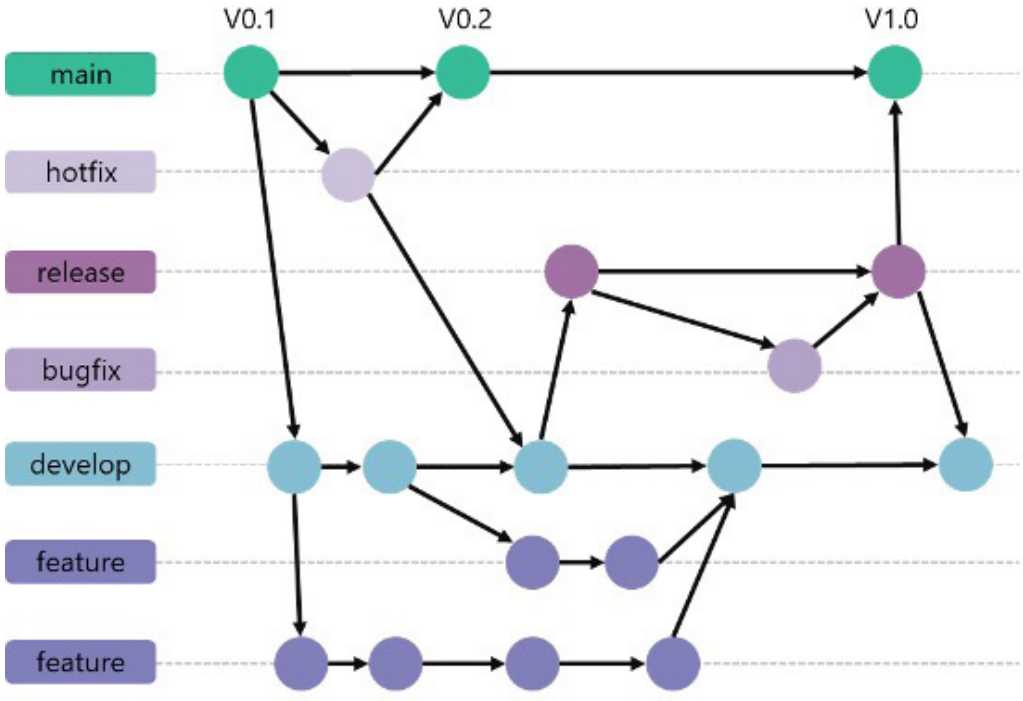
Fuente: DANE.

c) Utilizar un modelo de ramificación como guía para el flujo de trabajo

Implementar un modelo de ramificación es crucial para establecer un flujo de trabajo estructurado en la gestión de ramas, proporcionando una jerarquización clara que optimiza el proceso de desarrollo colaborativo. En esta guía se sugiere seguir el modelo Gitflow de (Wachsman, 2015). Este enfoque (Figura 6) es ampliamente adoptado en proyectos de ciencia de datos debido a su capacidad para integrarse de manera óptima con los paradigmas de Integración Continua (CI) y Despliegue Continuo (CD). Este funciona así:

- **Crear ramas históricas:** en lugar de tener una sola rama *(main/master)* utiliza la rama de *develop* como rama de integración para las funciones.
- **Crear ramas de características:** crea una rama para cada nueva característica ramificada y fusionada con la rama *develop.*
- **Crear ramas de liberación:** una vez que *develop* haya adquirido suficientes características para un lanzamiento se debe bifurcar una rama de *develop* llamada *release* y una vez esté lista fusione con *(main/master).*
- **Crear ramas de mantenimiento:** existen dos tipos de ramas de mantenimiento las de corrección critica *(hotfix)* y las de corrección de errores en desarrollo *(bugfix).* Las *hotfix* se bifurca directamente de *main/master* y tan pronto como se complete la corrección se deben fusionar con *main/master* y *develop.* Mientras que las *bugfix* se bifurcan de la rama *release* y se una vez se corrigen se fusiona a la misma.

Figura 6. Flujo de trabajo de ramas en un repositorio



Fuente: (Wachsman, 2015)

d) Estandarizar los mensajes de confirmación (commits)

La estandarización de los mensajes de *commits* es fundamental para garantizar la claridad y la coherencia en el historial de un proyecto de desarrollo. Esto facilita la revisión del código al proporcionar una estructura consistente y fácilmente entendible. Se sugiere seguir la siguiente convención:

<id opcional> <**tipo**>: ([**alcance opcional**]) : <**Descripción**>
[**cuerpo opcional**]
[**pie de página opcional**]

---

feat:(**data augmentation**): **Implementar pipeline de data augmentation para clasificación de imágenes**
Se agregó un pipeline de data augmentation para mejorar el conjunto de datos de entrenamiento para el modelo de clasificación de imágenes.
Revisado por: Alejandro <asandovalp@example.com>

Tipo: indica el cambio incluido en la confirmación y puede ser:

- **feat:** para una nueva característica o funcionalidad.
- **fix:** para corrección de errores.
- **docs:** para cambios en la documentación.
- **style:** para cambios en el formato del código (sin afectar la funcionalidad).
- **refactor:** para cambios en el código que no son correcciones ni características nuevas.
- **test:** para adición o modificación de pruebas.
- **chore:** para tareas menores que no encajan en las otras categorías (por ejemplo, configuración del entorno).

Alcance: se puede agregar un alcance al tipo de una confirmación para ofrecer información contextual adicional, y se incluye entre paréntesis.

Descripción: se puede agregar un resumen breve y preciso del cambio realizado. Debe limitarse a 50 caracteres como máximo para garantizar claridad y evitar truncamientos. De ser necesario, se pueden agregar detalles adicionales en el cuerpo del mensaje del commit debajo de la línea de descripción.

Cuerpo: se puede agregar un cuerpo al mensaje para incluir explicaciones detalladas en la confirmación. Se sugiere que este no exceda los 72 caracteres.

Pie de página: se utiliza un pie de página para transmitir información adicional sobre la confirmación, como revisado por, aprobado por, etc.

Nota: *si se tiene una herramienta de gestión de proyectos, se debería mencionar entonces el id del issue/feature, o caso de uso, o historia de usuario, o id de requerimiento. Esto permite la trazabilidad entre el commit y los requerimientos.*

e) Verificar que todo este correcto antes de solicitar el merge request

Antes de solicitar la unión de ramas *(merge request - MR),* es esencial verificar que todos los cambios propuestos estén correctos y listos para ser integrados. Este proceso minimiza el riesgo de conflictos y asegura una integración fluida en el ciclo de vida del desarrollo. Para ello se recomienda:

- **Sincronizar la rama de trabajo con el main:** se debe asegurar de que su rama esté actualizada con respecto a la rama base (*main, develop,* o cualquier rama de integración). Esto ayuda a prevenir conflictos y asegurar que se está trabajando sobre la versión más reciente del código. Para ello, se debe realizar un git fetch seguido de un git rebase o git merge desde la rama *main* hacia su rama de trabajo.

- **Revisar los cambios que se van a subir:** se debe inspeccionar los cambios en su rama para asegurarse de que solo las modificaciones necesarias estén incluidas y que no se introduzcan cambios no deseados. Para ello, se debe usar git diff para revisar las diferencias y confirmar que los cambios son correctos y relevantes.

- **Ejecutar pruebas básicas y validaciones:** se deben realizar pruebas unitarias y funcionales para verificar que su código no introduzca errores o fallos. Debe asegurarse de que las pruebas pasen en su entorno local y, si es posible, en un entorno de integración continua (CI). Para ello, se debe ejecutar el conjunto de pruebas automatizadas configuradas para el proyecto (por ejemplo, usando pytest, JUnit, Jest).

- **Solicita el MR:** una vez que todos los pasos anteriores se han completado, se debe solicitar el merge request para integrar sus cambios en la rama base. Además, se debe proporcionar una descripción detallada de los cambios y cualquier contexto relevante.

f) Revisar y aprobar el código de forma ordenada

Se debe implementa un proceso de revisión de código donde otros miembros del equipo revisan y aprueban los cambios antes de fusionarlos. Esto ayuda a mejorar la calidad del código y detectar problemas antes de que lleguen a la producción. Para ello se deben crear *pull requests* para solicitar revisiones de código y asegurarse de que al menos un revisor apruebe los cambios.

g) Generar estrategias de resolución de conflictos

Se deben adoptar estrategias para resolver conflictos de manera efectiva cuando se presentan durante el proceso de fusión. Esto incluye herramientas y métodos para resolver conflictos de manera eficiente. Se pueden usar herramientas de fusión, como git mergetool, y seguir un enfoque sistemático para resolver conflictos.

4.
APLICACIÓN **DE LA GUÍA EN UN CASO DE PRUEBA**

Con el fin de demostrar la aplicabilidad de la presente guía, se ha implementado un caso de prueba real que incorpora todas las fases clave del ciclo de vida de un modelo de ciencia de datos antes descritas. Se espera que este caso práctico sirva como un ejemplo detallado y contextualizado de cómo aplicar las directrices de la guía en situaciones reales, proporcionando una base sólida para su replicación y adaptación en proyectos similares dentro del ámbito de la ciencia de datos.

El caso de prueba real se centra en la gestión, el despliegue y el mantenimiento de un modelo de predicción de sexo basado en los nombres de personas en español, utilizando una arquitectura en Azure Cloud. Esta elección se justifica por las capacidades de alta disponibilidad, escalabilidad automática y elasticidad que ofrece Azure, esenciales para un despliegue eficiente y adaptable a la demanda. Además, Azure facilita un entorno colaborativo unificado, donde equipos de desarrollo, operaciones y ciencia de datos pueden trabajar de manera integrada, optimizando el ciclo de vida del modelo desde la experimentación hasta la implementación y monitoreo continuo.

Para este caso puntual, el proceso comienza con la ingesta de datos mediante Azure Data Factory, asegurando la preparación y disponibilidad de los datos. Luego, Azure Batch se encarga del procesamiento paralelo de los datos, optimizando su calidad antes de ser utilizados en el entrenamiento. La experimentación y el entrenamiento del modelo se realizan en Azure Databricks, con gestión de versiones y experimentos a través de MLflow, mientras que los artefactos se almacenan en Azure Blob Storage. GitLab facilita la integración y el despliegue continuo (CI/CD), automatizando la implementación del modelo en un endpoint escalable utilizando Azure Kubernetes Service (AKS). El backlog y la planificación del proyecto son gestionados en Azure DevOps, garantizando un desarrollo ágil y estructurado. Finalmente, Azure Monitor se integra para supervisar constantemente las métricas de performance del modelo y los recursos de cómputo, asegurando un rendimiento óptimo en producción.

A continuación, se detallan de manera organizada los componentes clave de los procesos críticos descritos en esta guía, aplicados en el caso de prueba.

Inicio del proyecto de ciencia de datos

1

Análisis del problema empresarial

Roles que intervienen

S AN*

Actividades: evaluación detallada para identificar el problema y definir los objetivo y métricas del negocio. Propuesta de la hoja de ruta y características del producto para resolver el problema.

Resultado:

¿Cuál es el problema empresarial que pretende resolver?
R/ Enriquecer los registros administrativos que contienen dentro de sus columnas la variable de nombres y añadir la variable cualitativa de sexo en aquellos casos donde esta información no está presente.

¿Cuál es el objetivo principal del proyecto desde la perspectiva empresarial?
R/ Contar con un modelo de aprendizaje profundo para predecir el sexo de una persona a partir de su nombre en español con el fin de que sirva como método de enriquecimiento de datos en registros administrativos.

¿Quiénes son los principales stakeholders y usuarios finales del proyecto, y cuáles son sus necesidades y expectativas?
R/ Usuarios temáticos del Grupo Interno de Trabajo de Prospectiva y Analítica de datos de la Dirección de Regulación, Planeación, Estandarización y Normalización, quienes esperan contar con una solución capaz de enriquecer registros administrativos y otros conjuntos de datos en los que la variable sexo no esté presente.

¿Cuál es la justificación para optar por una solución de ciencia de datos?
R/ Automatizar el enriquecimiento de datos asegurando la consistencia en los resultados y minimizando errores humanos.

¿Cuáles son los criterios y las métricas que indicarán si la solución de ciencia de datos ha logrado los resultados esperados?
R/ Desarrollar un modelo entrenado que alcance una curva ROC con un área bajo la curva (AUC) superior al 90% y una precisión (accuracy) superior al 90% en sus predicciones sobre datos no vistos.

¿Qué impacto se espera que tenga la solución de ciencia de datos en el negocio, y cómo se integrará en los procesos existentes?
R/ Se espera que la solución propuesta para enriquecer los registros administrativos incremente significativamente la utilidad de los datos, habilitando análisis avanzados que incorporen la variable de sexo. La automatización de este proceso garantizará una actualización continua y consistente de la información, elevando el nivel de completitud de los datos.

¿Qué fechas límite se han establecido para la finalización del proyecto y qué debe incluirse en cada entregable para cumplir con las expectativas del cliente?
R/ Se tiene como fecha límite los últimos días de septiembre y cada entregable se presenta en la última semana de cada mes siguiendo las etapas expuestas en el cronograma de actividades.

Punto de control: con base en las respuestas de los stakeholders se establece que el equipo requiere de por lo menos un líder de proyecto, arquitecto de soluciones, ingeniero de datos, científico de dato, ingeniero de machine learning y desarrollador de aplicaciones.

Artefactos

Secciones diligenciadas del Documento Marco (project_charter.md)
- Nombre del proyecto
- Objetivo del proyecto
- Alcance del proyecto

Infraestructura, herramientas y utilidades

- Metodología
- Cronograma
- Equipo del proyecto
- Stakeholders
- Aprobaciones

Azure DevOps
(Gestión del backlog del proyecto)

GitLab
(Documentación del proyecto)

Inicio del proyecto de ciencia de datos

2

Derivar el objetivo empresarial en un problema de ciencia de datos

Actividades: traducción de los objetivos del negocio en problemas específicos de ciencia de datos. Investigación y evaluación del estado actual de los modelos dentro de la organización o de los avances en la industria, alineados con las necesidades estratégicas y los objetivos del negocio.

Resultado:

El AN* en conjunto con el CD identificaron que existe un estudio que aborda un problema similar al que se está trabajando, en el cual se predice el sexo de personas utilizando nombres en portugués de Brasil mediante redes BiLSTM. Este estudio alcanzó una precisión (accuracy) del 96%, estableciendo una métrica base y un objetivo clave para el éxito de nuestro proyecto. Este resultado demuestra la viabilidad del enfoque.

Se establecen como umbrales o criterios de éxito un modelo entrenado que alcance una curva ROC con un área bajo la curva (AUC) superior al 90% y una precisión (accuracy) superior al 90% en sus predicciones sobre datos no vistos.

Secciones diligenciadas del Documento Marco (project_charter.md)
- Alcance (Criterios de éxito del proyecto)

Roles que intervienen

Artefactos

Azure DevOps
(Gestión del backlog del proyecto)

GitLab
(Documentación del proyecto)

Infraestructura, herramientas y utilidades

Inicio del proyecto de ciencia de datos

3

Entender los datos necesarios para resolver el problema

Roles que intervienen

Actividades: identificación y evaluación de los datos requeridos para abordar el problema: disponibilidad, accesibilidad y categorías de estos.

Resultado:

¿Qué tipos de datos están disponibles para abordar el problema empresarial?
R/ Se dispone de un conjunto de datos que contiene nombres de personas en español, etiquetados por sexo, con una distribución balanceada de aproximadamente 25.000 registros únicos para cada categoría de mujeres y hombres disponibles abiertamente en la plataforma de Kaggle. Además, se cuenta con la base de datos histórica de SISBEN IV, que también está etiquetada con la variable de sexo, proporcionando una fuente adicional de datos reales y relevantes para el entrenamiento y la evaluación de los modelos.

¿Dónde se almacenan los datos y qué sistemas o bases de datos se utilizan para su gestión?
R/ Los datos se almacenarán en Azure Blob Storage.

¿Qué tamaño tienen los conjuntos de datos y en qué formatos están almacenados?
R/ El conjuntos de datos de Kaggle se encuentran en dos archivos .csv, con un peso aproximado de 800 Kb. Mientras que el de SISBEN IV se encuentra en formato .txt con un tamaño aproximado de 500 KB.

¿Qué nivel de confidencialidad o sensibilidad tienen los datos? ¿Se requieren procesos de cifrado?

R/ Si bien los conjuntos de datos incluyen una variable considerada sensible, la ausencia de apellidos y de información adicional impide la identificación directa o indirecta de los individuos. Por lo tanto, no es necesario implementar procesos de cifrado para proteger la identidad de los sujetos.

¿Cómo se recopilan los datos y con qué frecuencia se actualizan o enriquecen?
R/ Los datos fueron recopilados desde fuentes públicas y bases de datos institucionales, como Kaggle y SISBEN IV. Los datos de Kaggle son estáticos y no se actualizan regularmente, mientras que la base de datos SISBEN IV se actualiza diariamente como parte de los procesos de recolección de información del gobierno.

¿Cuál es el nivel de calidad de los datos disponibles y qué problemas han identificado?
R/ En cuanto a la base de datos SISBEN IV, aunque es una fuente confiable y actualizada, existen problemas relacionados con registros duplicados, errores ortográficos, y desnormalización, derivados del proceso de recolección del manual de los datos.

Punto de control: con base en las respuestas, el ID y CD establecen que el proyecto es técnicamente viable y no presenta limitaciones que generen alertas para su desarrollo y se lo comunican al AN*.

Artefactos

Secciones diligenciadas del Documento Marco (project_charter.md)
- Alcance (Descripción de los datos disponibles)

Infraestructura, herramientas y utilidades

Azure DevOps
(Gestión del backlog del proyecto)

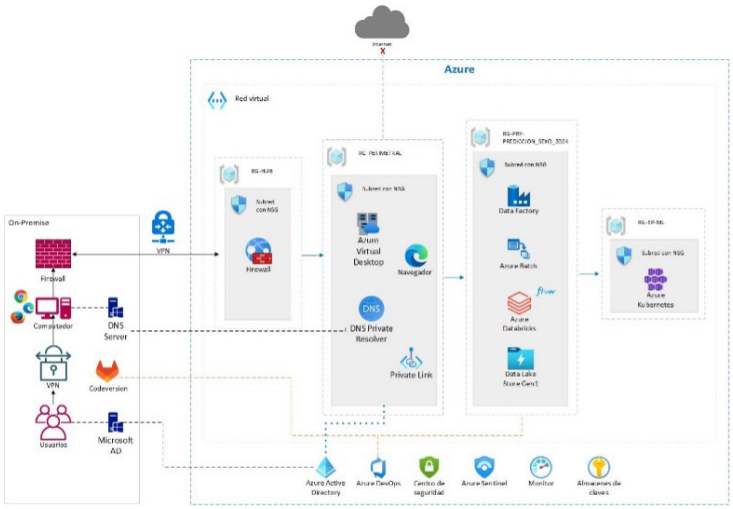
GitLab
(Documentación del proyecto)

Inicio del proyecto de ciencia de datos

Actividades: planificación de la infraestructura y la selección de tecnologías adecuadas para el proyecto.

Resultado:

El AS diseño la siguiente arquitectura de solución:



El costo aproximado proyectado para mantener la solución de ciencia de datos en producción, incluyendo el talento humano y la infraestructura es de aproximadamente $75.000.000 COP.

Punto de control: con base en el presupuesto establecido, el AN* determino que proyecto es viable desde el punto de vista financiero.

Secciones diligenciadas del Documento Marco (project_charter.md)

- Arquitectura de la solución
- Archivo de configuración del proyecto y sus dependencias (pyproject.toml)
- Archivo markdown de descripción del proyecto (README.md)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab
(Documentación del proyecto)

4

Diseño de la arquitectura y tecnologías a utilizar

Roles que intervienen

S AN* AS

Artefactos

Infraestructura, herramientas y utilidades

Inicio del proyecto de ciencia de datos

5

Definir el protocolo de acceso de conexión a los datos

Roles que intervienen

Artefactos

Infraestructura, herramientas y utilidades

Actividades: identificar la localización de los datos y definir el protocolo de acceso.

Resultado:

El S indica que las fuentes del origen de los datos están en dos ubicaciones:

- Página web de Kaggle: https://www.kaggle.com/datasets/migalpha/spanish-names?resource=download
- Base de datos Oracle alojada en los servidores locales del DANE (Indicando los parámetros de conexión de esquema, contraseña, host, puerto, nombre del servicio y nombre de tabla).

Secciones diligenciadas del Documento Marco (project_charter.md)

- Alcance (Descripción de los datos disponibles)

Oracle Database
(Origen)

Azure Data Factory
(Ingesta de datos)

Azure Blob Storage
(Almacenamiento de datos)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab
(Documentación del proyecto)

Ingeniería de características

6

Conexión con los datos

Actividades: establecer el acceso y la conexión a las fuentes de datos en su estado original que pueden ser de tipo : en vivo (on-streaming), por lotes (batch) o alojados en algún sistema de almacenamiento de datos. En el caso donde la

fuente correspondiente y configurar la conexión necesaria para su integración.

Resultado:

El ID realiza las pruebas de conexión con los fuentes de origen de los datos y desarrolla funciones de conexión de los datos y las valida previo a ser incluidas en del pipeline de preprocesamiento de datos (data_pipeline.py).

Secciones diligenciadas del Informe resumen de datos (data_summary.md)
- Origen de los datos
- Seguridad y privacidad de datos
- Diccionario de datos

Script Python con funciones de conexión de los datos del pipeline de preprocesamiento de datos (data_pipeline.py) en código Python

Oracle Database
(Origen)

Azure Data Factory
(Ingesta de datos)

Azure Blob Storage
(Almacenamiento de datos)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab
(Documentación del proyecto)

Roles que intervienen

Artefactos

Infraestructura, herramientas y utilidades

Ingeniería de características

7

Extracción de datos

Roles que intervienen

Actividades: obtener los datos de las conexiones previamente definidas para su posterior análisis. Para aquellos proyectos que requieran enriquecer sus datos a partir de fuentes secundarias se consideran estrategias como Web scraping a servicios web, redes sociales, aplicaciones móviles, entre otros.

Resultado:

El ID desarrolla Script Python con funciones de conexión de los datos del pipeline de preprocesamiento de datos (data_pipeline.py), el cual se ejecuta en el servicio de Azure Data Factory, quien actúa como intermediario entre las fuentes de origen de datos y el almacenamiento de datos Azure Blob Storage.

Artefactos

Secciones diligenciadas del Informe resumen de datos (data_summary.md)
- Rutinas para la carga de datos

Script Python con funciones de extracción de los datos del pipeline de preprocesamiento de datos (data_pipeline.py) en código Python

Infraestructura, herramientas y utilidades

Azure Blob Storage
(Almacenamiento de datos)

Azure Batch
(Procesamiento de datos)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab
(Documentación del proyecto)

Ingeniería de características

8

Transformación y limpieza de datos

Roles que intervienen

ID

Actividades: procesamiento de datos para corregir errores, eliminar inconsistencias y formateo adecuado para garantizar la integridad de estos basado en un análisis preliminar exploratorio. Incluye la normalización, la agregación, y demás reglas establecidas para la limpieza de datos.

Resultado:

El ID definió como reglas de transformación:
- Codificación de la variable objetivo 0 y 1.
- Conversión a minúsculas.
- Concatenación de primer y segundo nombre.

El ID desarrolla Script Python con funciones de

transformación de datos incluyéndolas como procesos del pipeline de preprocesamiento de datos (data_pipeline.py) , las cuales se ejecutan en el servicio de Azure Batch.

Secciones diligenciadas del Informe resumen de datos (data_summary.md)
- Movimiento e ingeniería de datos
- Resumen de calidad de datos

Script Python de la preprocesamiento de datos (data_preprocessing.py)

Azure Blob Storage
(Almacenamiento de datos)

Azure Batch
(Procesamiento de datos)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab
(Documentación del proyecto)

Artefactos

Infraestructura, herramientas y utilidades

Ingeniería de características

9

Ingeniería de características

Roles que intervienen

ID CD AD

Actividades: creación de nuevas variables a partir de datos existentes, optimizando así la representación de los datos para mejorar el rendimiento del modelo. Incluye la selección de atributos relevantes que pueden aumentar la precisión predictiva y la eficiencia del modelo. Los roles que intervienen en esta tarea son principalmente: el ingeniero de datos y científico de datos.

Resultado:

el CD y AD en común acuerdo definieron la regla de ingeniería de características des:
- Tokenización de los nombres a nivel de carácter descomponiendo cada nombre en sus componentes individuales utilizando métodos nativos de Python.

El ID desarrolla Script Python con funciones de ingeniería de características incluyéndolas como

Artefactos

procesos del pipeline de preprocesamiento de datos (data_pipeline.py), las cuales se ejecutan en el servicio de Azure Batch.

Secciones diligenciadas del Informe resumen de datos (data_summary.md)
- Movimiento e ingeniería de datos

Script Python de la ingeniería de características (feature_engineering.py)

Infraestructura, herramientas y utilidades

Azure Blob Storage
(Almacenamiento de datos)

Azure Batch
(Procesamiento de datos)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab
v(Documentación del proyecto)

Ingeniería de características

Actividades: recopilación y carga de datos ya sea por lotes (batch) o en línea (on-streaming) en el sistema de almacenamiento de características y este puede ser una base de datos en línea o fuera de línea.

Resultado:

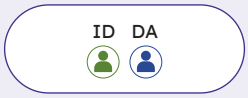
El ID desarrolla Script de Python del pipeline de preprocesamiento de datos (data_pipeline.py), el cual se ejecuta en el servicio de Azure Data Factory (ingesta de datos) conectado con Azure Blob Storage (almacenamiento de datos) y Azure Batch (procesamiento de datos). Y se comunica con el DA para integrar este pipeline en una orquestación de CI/CD.

Script Python del pipeline de preprocesamiento de datos (data_pipeline.py)

10

Trabajo de ingesta de datos

Roles que intervienen



Artefactos

Azure Data Factory
(Ingesta de datos)

Azure Blob Storage
(Almacenamiento de datos)

Azure Batch
(Procesamiento de datos)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab
(Documentación del proyecto)

Infraestructura, herramientas y utilidades

Experimentación

11

Análisis de datos

Roles que intervienen

ID AD CD

Artefactos

Infraestructura, herramientas y utilidades

Actividades: realizar el Análisis Exploratorio de datos (EDA) para identificar patrones, tendencias y relaciones entre diversas características y la variable objetivo. Esto incluye estadísticas descriptivas, visualización de datos, distribución y correlación de variables.

Resultado:

El AD y CD en apoyo del ID conectan servicio de Azure Blob Storage con Azure Databricks para realizar el EDA en un Notebook Python de preprocesamiento de datos (data_preprocessing.ipynb).

Secciones diligenciadas del Informe resumen de datos (data_summary.md)
- Análisis Exploratorio de Datos

Notebook Python de EDA (data_exploration.ipynb)

Azure Blob Storage
(Almacenamiento de datos)

Azure Databricks
(Experimentación y desarrollo del modelo)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab
(Documentación del proyecto)

Experimentación

12

Preparación y validación de datos

Actividades: preprocesamiento para asegurar un conjunto de datos con alta calidad antes del modelado. Incluye la detección de anomalías en los datos mediante la aplicación de reglas específicas alineadas con los requisitos del negocio como: Control sobre posibles desviaciones en la distribución estadística de los datos en el tiempo, cambios inesperados en correlaciones o patrones entre variables, valores faltantes, formatos inválidos y sesgo en los datos que puedan influir en los resultados de los modelos . También incluye la creación de conjuntos de datos divididos en entrenamiento, prueba y validación.

Resultado:

El CD elabora las reglas de preparación y validación de datos, que incluye la eliminación de los nombres con una longitud superior a 27 caracteres debido a las restricciones impuestas durante la configuración del modelo, específicamente en la capa de entrada de este. La tokenización a nivel de carácter con un límite de 27 caracteres fue seleccionada para equilibrar la complejidad computacional y la capacidad de capturar el contexto dentro de los nombres. Además, las desarrolla en el Notebook Python de preprocesamiento de datos (data_preprocessing.ipynb).

Roles que intervienen

CD

Artefactos

Notebook Python de preprocesamiento de datos (data_preprocessing.ipynb)

Infraestructura, herramientas y utilidades

Azure Databricks
(Experimentación y desarrollo del modelo)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab
(Documentación del proyecto)

Experimentación

13

Entrenamiento de modelos

Roles que intervienen

CD

Artefactos

Infraestructura, herramientas y utilidades

Actividades: aplicación de algoritmos de ciencia de datos utilizando los datos preprocesados en las etapas previas.
Nota: es recomendado utilizar el principio de parsimonia (navaja de OCCAM), comenzando con modelos simples y progresando hacia modelos más complejos. Se realiza el entrenamiento, ajuste de hiperparámetros y selección de mejores modelos.

Resultado:

El CD inicia los experimentos entrenando modelos LSTM y BiLSTM utilizando únicamente los datos de Kaggle realizando pruebas con los datos de SISBEN IV versionándolos con MLFlow y desarrollándolos en un Notebook Python de entrenamiento de modelos (model_training.ipynb).

Secciones diligenciadas del Informe resumen de modelos (model_summary.md)
- Descripción del problema
- Experimentación

Notebook Python de entrenamiento de modelos (model_training.ipynb)

Azure Databricks con MLFlow
(Experimentación y desarrollo del modelo)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab
(Documentación del proyecto)

Experimentación

14

Validación de modelos

Actividades: realizar la ingeniería de modelos para identificar el algoritmo y los hiperparámetros de mejor rendimiento para el modelo.

Resultado:

El CD experimentó diferentes parámetros del modelo de forma interactiva en el entrenamiento

del modelo hasta alcanzar las métricas de rendimiento previamente definidas como óptimas, en la cual se obtuvo un modelo BiLSTM con un accuracy de 0,9623 y ROC 0,96. El modelo se versiono con MLFlow y desarrollo en un Notebook Python de evaluación de modelos (model_evaluation.ipynb).

Secciones diligenciadas del Informe resumen de modelos (model_summary.md)
- Registros de entrenamiento
- Evaluación del modelo

Notebook Python de evaluación de modelos (model_evaluation.ipynb)
Modelos e hiper parámetros

Azure Databricks con MLFlow
(Experimentación y desarrollo del modelo)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab
(Documentación del proyecto)

Roles que intervienen

CD

Artefactos

Infraestructura, herramientas y utilidades

Experimentación

15

Exporte del modelo

Roles que intervienen

DA IML CD

Actividades: guardar y preparar en el repositorio corporativo el modelo entrenado que cumpla el rendimiento y las métricas de negocio establecidas en el primer proceso clave para su implementación en entornos de producción. Así como el código, los parámetros seleccionados y la documentación del modelo.

Resultado:

El CD exporta el modelo final e hiper parámetros (model.pkl) y envía el código al repositorio del proyecto.

Mientras que el IML define el código para la canalización del flujo de trabajo automatizado de ciencia de datos y lo envía al repositorio. Para que cada vez que el CD envíe un nuevo modelo o el IML envíe un nuevo código de canalización de flujo de trabajo al repositorio, el componente CI/CD detecte el código actualizado y active

automáticamente la canalización de CI/CD que lleva a cabo los pasos de compilación, prueba y entrega. El paso de compilación crea artefactos que contienen el modelo y las tareas de la canalización del flujo de trabajo. El paso de prueba valida los criterios de calidad seleccionados para la aceptación de la solución, el modelo y el código de canalización del flujo de trabajo. Mientras que el paso de entrega envía los artefactos versionados, al almacén de artefactos.

Artefactos

Secciones diligenciadas del Informe resumen de modelos (model_summary.md)
- Descripción del modelo final

Modelo final e hiper parámetros (model.pkl)

Infraestructura, herramientas y utilidades

Azure Databricks con MLFlow
(Experimentación y desarrollo del modelo)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab
(Documentación del proyecto)

Pipeline del flujo de trabajo automatizado de ciencia de datos y servicio y monitoreo del modelo

16

Extracción de datos

Actividades: extracción automatizada de las características versionadas en los sistemas de almacenamiento de características que pueden extraerse de la base de datos en línea o fuera de línea (o de cualquier tipo de almacén de datos).

Resultado:

El IML desarrolla el Script Python del preprocesamiento de datos (data_preprocessing.py) que incluye un programador (trigger) que detecta cuando hay nuevos datos disponibles basado en algún evento o programación periódica activando el pipeline del flujo de trabajo automatizado de ciencia de datos.

Roles que intervienen

Actividades: extracción automatizada de las características versionadas en los sistemas de almacenamiento de características que pueden extraerse de la base de datos en línea o fuera de línea (o de cualquier tipo de almacén de datos).

Resultado:

El IML desarrolla el Script Python del preprocesamiento de datos (data_preprocessing.py) que incluye un programador (trigger) que detecta cuando hay nuevos datos disponibles basado en algún evento o programación periódica activando el pipeline del flujo de trabajo automatizado de ciencia de datos.

Script Python del preprocesamiento de datos (data_preprocessing.py)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab CI/CD
(Documentación del proyecto y Pipeline de CI/vCD)

Artefactos

Infraestructura, herramientas y utilidades

Pipeline del flujo de trabajo automatizado de ciencia de datos y servicio y monitoreo del modelo

17

Preparación y validadición de datos

Roles que intervienen

IML

Artefactos

Actividades: preparación y validación automatizada de datos basada en las reglas definidas en la etapa de experimentación.

Resultado:

El IML desarrolla el Script Python del pipeline de preprocesamiento de datos (data_pipeline.py) e ingeniería de características (feature_engineering.py) que se disparan de forma automática una vez finaliza el proceso anterior de Extracción de datos automatizada.

Script Python de la ingeniería de características (feature_engineering.py)
Script Python del pipeline de preprocesamiento de datos (data_pipeline.py)

Infraestructura, herramientas y utilidades

Azure DevOps
(Gestión del backlog del proyecto)

GitLab CI/CD
(Documentación del proyecto y Pipeline de CI/CD)

Pipeline del flujo de trabajo automatizado de ciencia de datos y servicio y monitoreo del modelo

Actividades: entrenamiento automatizado del modelo final sobre nuevos datos no observados. El algoritmo y los hiperparámetros ya están predefinidos en función de la configuración de la etapa de experimentación anterior. El modelo es reentrando y refinado.

Resultado:

El IML desarrolla el Script de Python con métodos del modelo (model.py), Script de Python del pipeline de entrenamiento (training_pipeline.py) y Script Shell de entrenamiento (run_training.sh) que se disparan de forma automática una vez finaliza el proceso anterior de Preparación y validación automatizada de datos.

Secciones diligenciadas del Informe resumen de modelos (model_summary.md)
- Descripción del Problema
- Experimentación

Script Shell de entrenamiento (run_training.sh)
Script Python con métodos del modelo (model.py)
Script Python del pipeline de entrenamiento (training_pipeline.py)

Azure Databricks con MLFlow
(Reentrenamiento del modelo)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab CI/CD
(Documentación del proyecto y Pipeline de CI/CD)

18

Entrenamiento de modelos / refinamiento

Roles que intervienen

IML

Artefactos

Infraestructura, herramientas y utilidades

Pipeline del flujo de trabajo automatizado de ciencia de datos y servicio y monitoreo del modelo

19

Validación de modelos

Roles que intervienen

IML

Actividades: realizar la ingeniería de modelos de forma automatizada para identificar el algoritmo y los hiperparámetros de mejor rendimiento para el modelo.

Resultado:

El IML desarrolla el Script de Python del pipeline de entrenamiento (training_pipeline.py) y Script Shell de entrenamiento (run_training.sh) que se disparan de forma automática una vez finaliza el proceso anterior de Entrenamiento y refinamiento automatizado del modelo de datos.

Artefactos

Secciones diligenciadas del Informe resumen de modelos (model_summary.md)
- Registros de entrenamiento
- Evaluación del modelo

Modelos e hiper parámetros
Script Shell de entrenamiento (run_training.sh)
Script Python del pipeline de entrenamiento (training_pipeline.py)

Infraestructura, herramientas y utilidades

Azure Databricks con MLFlow
(Reentrenamiento del modelo)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab CI/CD
(Documentación del proyecto y Pipeline de CI/CD)

Pipeline del flujo de trabajo automatizado de ciencia de datos y servicio y monitoreo del modelo

20

Exporte del modelo

Actividades: guardar y preparar automáticamente en el repositorio corporativo el modelo entrenado que cumpla el rendimiento y las métricas de negocio establecidas en el primer proceso clave para su implementación en entornos de producción. Así como el código y los parámetros seleccionados.

Resultado:

En el Script de Python del pipeline de entrenamiento (training_pipeline.py) y Script Shell de entrenamiento (run_training.sh) desarrollado por el IML una vez se cuenta con un modelo que alcanzó las métricas de rendimiento previamente definidas como óptimas, tanto a nivel de eficacia del modelo como de eficiencia computacional, se exporta el modelo final e hiper parámetros (model.pkl) de forma automatizada, enviándolo junto con el código al repositorio del proyecto.

Secciones diligenciadas del Informe resumen de modelos (model_summary.md)
- Descripción del modelo final

Modelo final e hiper parámetros (model.pkl)

Azure Databricks con MLFlow
(Reentrenamiento del modelo)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab CI/CD
(Documentación del proyecto y Pipeline de CI/CD)

Azure Blob Storage
(Almacenamiento de modelos)

Roles que intervienen

IML

Artefactos

Infraestructura, herramientas y utilidades

Pipeline del flujo de trabajo automatizado de ciencia de datos y servicio y monitoreo del modelo

21

Registro en el repositorio de modelos

Roles que intervienen

DA IML CD

Actividades: registro del modelo y sus artefactos en el almacén de metadatos de ML. Esto también se denomina registro del linaje de modelos el cual combina el versionado de datos y de código para cada modelo registrado, así como del estado del modelo: en preparación (staging) o listo para producción (production-ready).

Resultado:

Una vez finalizada la etapa 20. Exporte del modelo, el modelo (model.pkl) junto con sus hiper

parámetros y artefactos es registrado y almacenado en Azure Blob Storage de forma automática generando una nueva versión y cambiado su estado a listo para producción (production-ready).

Artefactos

Secciones diligenciadas del Informe resumen de despliegue (deployment_summary.md)
- Detalles del Modelo

Modelos e hiper parámetros (model.pkl), metadatos del entrenamiento del modelo y estados del modelo.

Infraestructura, herramientas y utilidades

Azure Blob Storage
(Almacenamiento de modelos junto con sus artefactos)

Pipeline del flujo de trabajo automatizado de ciencia de datos y servicio y monitoreo del modelo

22

Paso a producción

Roles que intervienen

DA IML CD

Actividades: preparar la canalización de integración y despliegue continuo que extraiga, construya, pruebe y despliegue el modelo y el código de servicio del modelo en producción. Esto debe incluir una estrategia de despliegue acorde con la infraestructura definida (por ejemplo, despliegue total o basado en canarios).

Resultado:

Una vez finaliza la etapa 21. Registro en el repositorio, el modelo es entregado de forma automática al IML para su implementación en un endpoint escalable utilizando Azure Kubernetes Service (AKS). Se activa el proceso de integración y despliegue continuo del componente de CI/CD utilizando GitLab CI/CD, en el cual la canalización de integración y despliegue continuo realiza la construcción y prueba del modelo y código de servicio, y lo despliega para su uso en producción. Para ello el IML debió desarrollar previamente los Scripts Shell de pruebas unitarias (run_tests.sh), Archivo de configuración de GitLab CI/CD (.gitlab-ci.yml), Script Python de pruebas unitarias de preprocesamiento de da-

tos (test_data_preprocessing.py), Script Python de pruebas unitarias de ingeniería de características (test_feature_engineering.py), Script Python de pruebas unitarias de modelos (test_model.py), Script Shell de despliegue (run_deployment.sh) y Script Python de despliegue (deploy_model.py) utilizando GitLab CI/CD.

Secciones diligenciadas del Informe resumen de despliegue (deployment_summary.md)
- Plataforma de despliegue
- Requisitos técnicos
- Requisitos de seguridad
- Etapas de liberación
- Diagrama de arquitectura
- Código de despliegue

Artefactos

Script Shell de pruebas unitarias (run_tests.sh)
Archivo de configuración de GitLab CI/CD (.gitlab-ci.yml)
Script Python de pruebas unitarias de preprocesamiento de datos (test_data_preprocessing.py)
Script Python de pruebas unitarias de ingeniería de características (test_feature_engineering.py)
Script Python de pruebas unitarias de modelos (test_model.py)
Script Shell de despliegue (run_deployment.sh)
Script Python de despliegue (deploy_model.py)

Infraestructura, herramientas y utilidades

Azure DevOps
(Gestión del backlog del proyecto)

GitLab CI/CD
(Documentación del proyecto y Pipeline de CI/CD)

Azure Kubernetes Services
(Despliegue del modelo)

Pipeline del flujo de trabajo automatizado de ciencia de datos y servicio y monitoreo del modelo

23

Operacionalización del modelo

Roles que intervienen

DA IML

Artefactos

Infraestructura, herramientas y utilidades

Actividades: gestionar el despliegue asegurando un rendimiento óptimo y permitiendo ajustes o reversiones según sea necesario. Para ello se debe realizar de forma continua y transversal la evaluación de: Integración, rendimiento y robustez de los modelo, así como a la implementación de pruebas de seguridad (vulnerabilidades y privacidad).

Resultado:

Una vez finaliza la etapa 22. Paso a producción, se activa el componente de servicio, el modelo está en la capacidad de generar predicciones a través de un llamado de un API REST ya sea con datos nuevos o no observados provenientes del sistema de almacenamiento de características que son por lotes (batches). Para ello el DA en colaboración del IML debieron desarrollar los Script Shell de generación de predicciones (run_prediction.sh) y Script de Python con el pipeline de predicción (prediction_pipeline.py) utilizando el servicio de Azure Kubernetes Services (Despliegue del modelo).

Secciones diligenciadas del Informe resumen de despliegue (deployment_summary.md)

- Monitoreo y operación del modelo

Script Shell de generación de predicciones (run_prediction.sh)
Script de Python con el pipeline de predicción (prediction_pipeline.py)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab CI/CD
(Documentación del proyecto y Pipeline de CI/CD)

Azure Kubernetes Services
(Despliegue del modelo)

Pipeline del flujo de trabajo automatizado de ciencia de datos y servicio y monitoreo del modelo

24
Monitoreo del modelo

Roles que intervienen

Artefactos

Infraestructura, herramientas y utilidades

Actividades: supervisar de manera continua el rendimiento de la solución, incluyendo la salud del sistema, detección de errores, latencias, métricas del modelo, garantizando así el funcionamiento óptimo y la calidad de la solución desplegada.

Resultado:

El componente de monitoreo (previamente configurado por el IML, DA y CD) y desarrollado en Script de Python de monitoreo del modelo (monitor_model.py) se encuentra activo continuamente en el servicio de Azure Monitor, es el que se encarga de observar de forma continua el rendimiento del servicio del modelo y la infraestructura en tiempo real. Si este detecta una degradación en las métricas de rendimiento, la información se transmite a través del bucle de retroalimentación, habilitando el reentrenamiento y mejora del modelo, hace que la información se transfiera a varios puntos receptores, como la etapa experimental, el pipeline de ingeniería de datos y el programador. La retroalimentación a la etapa experimental es gestionada por el CD para mejoras adicionales del modelo. La retroalimentación a la zona de ingeniería de datos permite el ajuste de las características preparadas para el sistema de almacenamiento de características.

Secciones diligenciadas del Informe resumen de despliegue (deployment_summary.md)
- Monitoreo y operación del modelo

Script de Python de monitoreo del modelo (monitor_model.py)

Azure DevOps
(Gestión del backlog del proyecto)

GitLab CI/CD
(Documentación del proyecto y Pipeline de CI/CD)

Azure Kubernetes Services
(Despliegue del modelo)

Azure Monitor
(Monitoreo y mantenimiento del modelo)

ANEXOS

Anexo A.	Documento marco (project_charter.md)
	Ver en: \pry-name-yyyy\assets\docs\project_charter.md
Anexo B.	Informe resumen de datos (data_summary.md)
	Ver en: \pry-name-yyyy\assets\docs\data_summary.md
Anexo C.	Informe resumen de modelos (model_summary.md)
	Ver: \pry-name-yyyy\assets\docs\model_summary.md
Anexo D.	Informe resumen de despliegue (deployment_summary.md)
	Ver en: \pry-name-yyyy\assets\docs\deployment_summary.md

REFERENCIAS

Agile Alliance. (2001). Agile Manifesto. https://agilemanifesto.org/iso/es/manifesto.html

Amazon Web Services. (2024a). Amazon EMR. https://aws.amazon.com/emr/

Amazon Web Services. (2024b). Amazon Kinesis. https://aws.amazon.com/es/kinesis/

Amazon Web Services. (2024c). Amazon QuickSight: Inteligencia empresarial unificada de Amazon QuickSight a hiperescala. https://aws.amazon.com/quicksight/

Amazon Web Services. (2024d). Amazon Redshift Spectrum. https://aws.amazon.com/redshift/spectrum/

Amazon Web Services. (2024e). Amazon S3. https://aws.amazon.com/es/s3/

Amazon Web Services. (2024f). Amazon SageMaker. https://aws.amazon.com/es/sagemaker/

Amazon Web Services. (2024g). Amazon SageMaker Autopilot. https://aws.amazon.com/sagemaker/autopilot/

Amazon Web Services. (2024h). AWS CodePipeline Documentation. https://docs.aws.amazon.com/codepipeline

Amazon Web Services. (2024i). Canary deployment: The slow rollout of a new version of an existing application. https://wa.aws.amazon.com/wellarchitected/2020-07-02T19-33-23/wat.concept.canary-deployment.en.html

Amazon Web Services. (2024j). DynamoDB Documentation. https://docs.aws.amazon.com/dynamodb

Anderson, D., & Reinertsen, D. (2010). Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press.

Ansible Community & Red Hat. (2024). How Ansible works. https://www.ansible.com/overview/how-ansible-works

Apache Software Foundation. (2024a). Apache Flink. https://flink.apache.org/

Apache Software Foundation. (2024b). Apache HBase Overview. https://hbase.apache.org/

Apache Software Foundation. (2024c). Apache Kafka. https://kafka.apache.org/

Apache Software Foundation. (2024d). HDFS Architecture Guide. https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

Apache Software Foundation. (2024e). PySpark Documentation. https://spark.apache.org/docs/latest/api/python/

Apache Software Foundation. (2024f). What is AirflowTM? https://airflow.apache.org/docs/apache-airflow/stable/index.html

Apache Software Foundation. (2024g). What is Apache Cassandra? https://cassandra.apache.org/doc/latest/

Apache Software Foundation. (2024h). What is Apache NiFi? https://nifi.apache.org/documentation/v2/

Apache Software Foundation. (2024i). What is Apache Spark? https://spark.apache.org/

Atlassian. (2024a). Bamboo documentation. https://confluence.atlassian.com/bamboo

Atlassian. (2024b). Bitbucket: Git solution for teams using jira. https://bitbucket.org/product

Atlassian. (2024c). Confluence Cloud resources. https://support.atlassian.com/confluence-cloud/resources/

Atlassian. (2024d). Jira Cloud resources. https://www.atlassian.com/software/jira

Atlassian. (2024e). Trello: Project Management Tool. https://trello.com

Atlassian. (2024f). What is a sprint backlog template? https://www.atlassian.com/software/jira/templates/sprint-backlog

Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A Software Architect's Perspective (Addison-Wesley Professional (ed.)).

Bicking, I. (2008). Pip [Software]. https://pip.pypa.io/

Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python (O'Reilly Media (ed.)).

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

Bokeh Contributors. (2024). Bokeh documentation. https://docs.bokeh.org/en/latest/

Bostock, M. (2024). D3.js - Data-Driven Documents. https://d3js.org

Bostrom, N. (2014). Superintelligence: Paths, dangers, strategies (Oxford University Press (ed.)).

Chen, T. (2023). XGBoost Documentation. https://xgboost.readthedocs.io/

Chollet, F. (2015). Keras: The Python Deep Learning API. https://keras.io/

CircleCI. (2024). CircleCI. https://circleci.com/

Cloud Native Computing Foundation. (2024). What is Kubernetes? https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

CollabNet. (2024). Apache Subversion: Enterprise-class centralized version control for the masses. https://subversion.apache.org/

Collibra Incorporation. (2024). Data Governance. https://www.collibra.com/us/en/products/data-governance

Collier, K. (2011). Agile Analytics: A Value-Driven Approach to Business Intelligence and Data Warehousing (Addison-Wesley Professional (ed.)).

Couchbase Inc. (2024). Couchbase Documentation. https://docs.couchbase.com/

Cournapeau, D., Grisel, O., Varoquaux, G., Gramfort, A., & Mueller, A. (2024). Scikit-learn: Machine Learning in Python. https://scikit-learn.org/

CRISP-DM Consortium. (1999). Cross-Industry Standard Process for Data Mining (CRISP-DM).

DataRobot. (2024). DataRobot Automated Machine Learning. https://www.datarobot.com/platform/ai-cloud/automated-machine-learning/

Docker Incorporated. (2024). What is Docker? https://www.docker.com/what-docker

DoltHub. (2024). What Is Dolt? https://docs.dolthub.com

Elementl. (2024). Dagster | Cloud-native orchestration of data pipelines. https://dagster.io/

Eustace, S. (2024). Poetry: Python dependency management and packaging made easy [Software]. https://github.com/python-poetry/poetry

Gift, N., & Deza, A. (2021). Practical MLOps: Operationalizing Machine Learning Models (O'Reilly Media (ed.)).

Gitea Community. (2024). Gitea. https://gitea.io/en-us/

Github Incorporated. (2024a). An open source Git extension for versioning large files. https://git-lfs.com/

Github Incorporated. (2024b). Github: Let's build from here. https://github.com/

Github Incorporated. (2024c). Repositories documentation. https://docs.github.com/es/repositories

Gitlab incorporated. (2024a). About GitLab. https://about.gitlab.com/

Gitlab incorporated. (2024b). Manage your code. https://docs.gitlab.com/ee/topics/manage_code.html

Goertzel, B. (2007). Artificial general intelligence: Concept, state of the art, and future prospects. Journal of Artificial General Intelligence, 1–48.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Google. (2024a). AutoML: Machine Learning Models Made Easy. https://cloud.google.com/automl

Google. (2024b). Cloud Storage Documentation. https://cloud.google.com/storage/docs

Google. (2024c). Dataflow. https://cloud.google.com/dataflow

Google. (2024d). Google Colab. https://colab.research.google.com/

Google. (2024e). Looker Studio docs. https://cloud.google.com/looker/docs?hl=es-419

Google. (2024f). What is Kubeflow? https://www.kubeflow.org/

Google Brain Team. (2024). An Open Source Machine Learning Framework for Everyone. https://www.tensorflow.org/

Google Cloud Platform. (2024). Vertex AI Documentation. https://cloud.google.com/vertex-ai/docs?hl=es-419

Grafana Labs. (2024). Grafana documentation. https://grafana.com/docs/grafana/latest/

H2O.ai. (2024). H2O AutoML. https://www.h2o.ai/products/h2o-automl/

HashiCorp. (2024a). Terraform by HashiCorp. https://www.terraform.io/

HashiCorp. (2024b). Vagrant Documentation. https://developer.hashicorp.com/vagrant/docs

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

Huang, X., Acero, A., Hon, H., & Reddy, R. (2001). Spoken Language Processing: A Guide to Theory, Algorithm, and System Development. In Prentice Hall.

Hugging Face. (2024). Accelerated Inference API. https://huggingface.co/docs/api-inference

IBM. (2024a). IBM Cloudant Documentation. https://cloud.ibm.com/docs/Cloudant

IBM. (2024b). IBM DB2 Documentation. https://www.ibm.com/docs/en/db2

Informatica. (2024). Data Governance & Compliance. https://www.informatica.com/products/data-governance/data-governance-and-compliance.html

Iterative.ai. (2024). Data Version Control. https://dvc.org/doc

JetBrains. (2024). PyCharm. https://www.jetbrains.com/pycharm/

Kawaguchi, K. (2024). Jenkins. https://www.jenkins.io/

Kreuzberger, D., Kuhl, N., & Hirschl, S. (2023). Machine Learning Operations (MLOps): Overview, Definition, and Architecture. IEEE Access, 11, 31866–31879. https://doi.org/10.1109/ACCESS.2023.3262138

Kuhn, M. (2024). The caret Package. https://cran.r-project.org/web/packages/caret/index.html

Lang, M. (2024). MLR3: A modern object-oriented machine learning framework in R. https://mlr3.mlr-org.com/

LangChain. (2024). LangChain Documentation. https://python.langchain.com/docs/

Legg, S., & Hutter, M. (2007). Universal Intelligence: A Definition of Machine Intelligence. ArXiv, 391–444.

Machine Learning for Developers. (2022). MLOps: MMLOps: Machine Learning Life Cycle. https://www.ml4devs.com/articles/mlops-machine-learning-life-cycle/

MariaDB Foundation. (2024). MariaDB Documentation. https://mariadb.com/kb/en/documentation/

Meta AI. (2016). PyTorch. https://pytorch.org/

Meyer, D. (2024). E1071: Misc functions of the Department of Statistics, Probability Theory Group. https://cran.r-project.org/web/packages/e1071/index.html

Microsoft. (2016). Team Data Science Process (TDSP).

Microsoft. (2024a). Azure Databricks. https://azure.microsoft.com/en-us/services/databricks/

Microsoft. (2024b). Azure DevOps Services. https://azure.microsoft.com/en-us/products/devops/

Microsoft. (2024c). Azure Pipelines documentation. https://learn.microsoft.com/en-us/azure/devops/pipelines/

Microsoft. (2024d). Introduction to Azure Storage. https://learn.microsoft.com/en-us/azure/storage/common/storage-introduction

Microsoft. (2024e). Unified Data Governance with Microsoft Purview. https://azure.microsoft.com/en-us/services/purview/

Microsoft. (2024f). Visual Studio Code. https://code.visualstudio.com/docs

Microsoft. (2024g). Welcome to Azure Stream Analytics. https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-introduction

Microsoft. (2024h). What is Azure Batch? https://docs.microsoft.com/en-us/azure/batch/batch-technical-overview

Microsoft. (2024i). What is Azure Data Lake Storage Gen2? https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction

Microsoft. (2024j). What is Azure Machine Learning? https://docs.microsoft.com/en-us/azure/machine-learning/overview-what-is-azure-machine-learning

Microsoft. (2024k). What is Power BI? https://powerbi.microsoft.com/en-us/what-is-power-bi/

Microsoft Corporation. (2024a). Azure Cosmos DB Documentation. https://learn.microsoft.com/en-us/azure/cosmos-db/

Microsoft Corporation. (2024b). SQL Server Documentation. https://docs.microsoft.com/en-us/sql/sql-server/

MLflow incorporated. (2024). MLflow: An open source platform for the machine learning lifecycle. https://mlflow.org/docs/latest/index.html

MongoDB Inc. (2024). MongoDB for GIANT Ideas. https://www.mongodb.com/what-is-mongodb

Moore, D. S., McCabe, G. P., & Craig, B. A. (2014). Introduction to the Practice of Statistics (W. H. Freeman (ed.)).

Neo4j Inc. (2024). Neo4j Documentation. https://neo4j.com/docs/

Notion Labs. (2024). Notion: La nueva generación de notas y documentos. https://www.notion.so/es-es/product/docs

OCDE. (2019). Principios de la IA Responsable. https://legalinstruments.oecd.org/en/instruments/OECD-LEGAL-0449

Oliphant, T. (2012). Anaconda [Software]. https://www.anaconda.com

OpenAI. (2024). Overview - OpenAI API. https://platform.openai.com/docs

OpenRefine. (2024). About OpenRefine. https://openrefine.org/

Oracle. (2024). Oracle Analytics. https://www.oracle.com/analytics/

Oracle Corporation. (2024a). MySQL. https://www.mysql.com/

Oracle Corporation. (2024b). Oracle Coherence Documentation. https://docs.oracle.com/en/middleware/standalone/coherence/

Oracle Corporation. (2024c). Oracle Database Documentation. https://docs.oracle.com/en/database/

Pachyderm Incorporated. (2024). Pachyderm: Automate data transformations. https://www.pachyderm.com/

Piatetsky-Shapiro, G. (1989). From Data Mining to Knowledge Discovery: An Overview. Advances in Knowledge Discovery and Data Mining, 1–34.

Plotly. (2024). Dash documentation & user guide: Plotly. https://plotly.com/dash/

Posit PBC. (2024a). RStudio. https://www.rstudio.com/

Posit PBC. (2024b). Shiny. https://shiny.rstudio.com/

PostgreSQL Global Development Group. (2024). About PostgreSQL. https://www.postgresql.org/about/

Prefect Technologies. (2024). Workflow Orchestration Made Simple | Prefect. https://www.prefect.io/

Project Jupyter. (2024). Project Jupyter. https://jupyter.org/

Provost, F., & Fawcett, T. (2013). Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking.

Pyenv Contributors. (2024). Pyenv: Simple Python version management [Software]. https://github.com/pyenv/pyenv

Quilt Data Incorporated. (2024). Introduction: Dev Quilt. https://docs.quiltdata.com/

R Core Team. (2024). The Comprehensive R Archive Network (CRAN). https://cran.r-project.org

RavenDB. (2024). RavenDB Documentation. https://ravendb.net/docs

Redis Labs. (2024). What is Redis? https://redis.io/docs/about/

Renv Contributors. (2024). Renv [Software]. https://github.com/rstudio/renv

Ridge, E. (2015). Guerrilla Analytics: A Practical Approach to Working with Data (1st editio).

RStudio. (2024a). Packrat. https://github.com/rstudio/packrat

RStudio. (2024b). R Markdown Formato for Flexible Dashboards - Flexdashbaord. https://rmarkdown.rstudio.com/flexdashboard/

Russell, S. J., & Norvig, P. (2016). Artificial Intelligence: A Modern Approach. Pearson.

Salesforce. (2024). What is Tableau? https://www.tableau.com/what-is-tableau

Sangani, S., Kalb, A., Nui, F., & Ganti, V. (2024). Alation: Data Governance. https://www.alation.com/data-governance/

SAS Institute. (2005). Sample, Explore, Modify, Model, Assess (SEMMA).

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks.

Schwaber, K., & Sutherland, J. (2020). The Definitive Guide to Scrum: The Rules of the Game. https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf

Seldon Technologies. (2024). Seldon Core: A software framework to deploy models into production. https://www.seldon.io/solutions/seldon-core

Spotify. (2024). Luigi. https://github.com/spotify/luigi

SQLite. (2024). SQLite Documentation. https://www.sqlite.org/docs.html

Stachowiak, H. (1973). General Model Theory (Springer (ed.)).

Superconductive. (2024). About Great Expectations. https://greatexpectations.io/

Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction (MIT Press (ed.)).

Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer.

Talend. (2024). Data Quality. https://www.talend.com/products/data-quality/

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. In: International conference on artificial neural networks. Springer, 9, 270.

Tecton.AI. (2024). Tecton: The Fastest Way to Build and Deploy Data Pipelines for Machine Learning. https://www.tecton.ai/product/

Tesseract OCR Team. (2020). Tesseract Open Source OCR Engine. GitHub. https://github.com/tesseract-ocr

Travis CI. (2024). Travis CI. https://travis-ci.com/

Treeverse Inc. (2024). Welcome to the Lake! https://docs.lakefs.io

Trifacta Incorporation. (2024). Data Preparation. https://www.alteryx.com/about-us/trifacta-is-now-alteryx-designer-cloud

Tukey, J. W. (1977). Exploratory Data Analysis. Addison-Wesley.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 5998–6008.

Wachsman, J. (2015). Git Flow A Successful Git Branching Model.

Weiss, K., Khoshgoftaar, T., & Wang, D. (2016). A survey of transfer learning. J Big Data, 3–9.

www.sen.gov.co

@DANE_Colombia

/DANEColombia

/DANEColombia

@DANEColombia