La Ciencia de Datos en la Institución

Estrategias para la integración del desarrollo y las operaciones

Clase 6

MAESTRÍA EN
SISTEMAS DE INFORMACIÓN
Mención Data Science

La excelencia no se improvisa

1. INTRODUCCIÓN DE LA CLASE

En este capítulo, exploramos las estrategias fundamentales para la integración de desarrollo y operaciones mediante las metodologías **DevOps** y **DataOps**, que se han vuelto esenciales en la gestión de proyectos de ciencia de datos. A medida que las organizaciones enfrentan la creciente demanda de datos precisos y procesados en tiempo real, DevOps y DataOps proporcionan un marco de referencia para optimizar cada fase del ciclo de vida de los datos y modelos. Estas metodologías no solo facilitan la colaboración entre equipos multidisciplinarios, sino que también impulsan la agilidad y la escalabilidad en el despliegue de soluciones analíticas.

DevOps se enfoca en mejorar el flujo de trabajo entre los equipos de desarrollo y operaciones mediante la automatización y la integración continua, mientras que DataOps aplica estos principios al contexto de la administración y análisis de datos. Al combinar ambos enfoques, las organizaciones pueden maximizar el valor de sus datos, minimizar errores y ofrecer soluciones adaptadas a las necesidades del mercado. En este capítulo, los estudiantes han analizado cómo implementar estos métodos y comprender el impacto positivo que tienen en la productividad y competitividad en diversos sectores.

Indicar los RDAs que corresponden a esta clase:

RDA 2: Aplicar marcos de referencia y metodologías para la gestión y gobierno de datos en ciencia de datos.

Clase 6: Estrategias para la integración del desarrollo y las operaciones

1.1. Introducción a DevOps: Integración de Desarrollo y Operaciones

DevOps integra desarrollo de software y operaciones para optimizar proyectos tecnológicos, mejorando la eficiencia y calidad en la entrega de servicios. Surge para adaptarse a entornos competitivos, donde la rapidez y flexibilidad son clave. Es esencial en tecnología y ciencia de datos, facilitando soluciones escalables y confiables. (Kim et al., 2016).

El concepto de DevOps se originó como una respuesta a los problemas de coordinación entre los equipos de desarrollo y operaciones en las organizaciones de tecnología. Tradicionalmente, estos equipos trabajaban de manera aislada: el equipo de desarrollo se enfocaba en crear nuevas funcionalidades o aplicaciones, mientras que el equipo de operaciones era responsable de mantener la infraestructura estable y segura. Esta separación, aunque permitía la especialización en cada área, resultaba en fricciones y demoras, especialmente cuando los desarrollos no se alineaban con las capacidades y limitaciones del entorno operativo (Forsgren et al., 2018). Para superar estos desafíos, DevOps propone unificar ambas áreas mediante principios de colaboración, automatización e integración continua.

Los principios clave de DevOps incluyen la automatización de procesos, donde se minimizan errores humanos y se aumenta la velocidad de despliegue automatizando tareas repetitivas en el desarrollo de software (Mishra & Sarawagi, 2019). La integración y entrega continua (CI/CD) permite integrar y probar código frecuentemente, detectando errores temprano y facilitando implementaciones rápidas y seguras en producción (Kim et al., 2016). Otro pilar es una cultura de colaboración y comunicación, que fomenta el trabajo conjunto entre desarrollo y operaciones, asegurando adaptabilidad y una rápida resolución de problemas en proyectos complejos (Erich et al., 2017). Además, DevOps resalta la importancia de una medición y monitoreo constante del rendimiento y estado de aplicaciones en cada fase, lo cual permite ajustes en tiempo real, asegurando confiabilidad y disponibilidad del servicio (Forsgren et al., 2018). Estos principios mejoran la eficiencia, adaptabilidad y calidad del ciclo de vida del desarrollo de software.

Enlace relacionado.

- ¿Qué es DevOps?
  Enlace:https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-devops
  En este enlace se explica las bases de DevOps, incluyendo sus principios y beneficios en la integración de equipos de desarrollo y operaciones. Es una introducción visual y práctica a las prácticas de automatización y colaboración en DevOps.

1.1.1. Beneficios de Implementar DevOps

DevOps aporta grandes beneficios a las organizaciones en velocidad, calidad y eficiencia. Al automatizar procesos e implementar la integración continua, DevOps reduce los tiempos de entrega y facilita la adaptación a los cambios de mercado. Según Forsgren et al. (2018), las organizaciones que adoptan DevOps pueden mejorar su rendimiento en un 50% en comparación con métodos tradicionales. Además, DevOps incrementa la calidad del software, ya que permite a los equipos comprender mejor las limitaciones de la infraestructura, logrando productos más robustos y adaptables (Kim et al., 2016). La entrega continua asegura versiones probadas y ajustadas, reduciendo problemas en producción. En términos de eficiencia, DevOps minimiza el trabajo manual, permitiendo que los

recursos se concentren en innovación. En ciencia de datos, donde los proyectos son iterativos y complejos, DevOps es crucial para implementar modelos y aplicaciones de forma ágil y sin interrupciones (Mishra & Sarawagi, 2019).

1.1.2. Herramientas Clave para Implementar DevOps

Para implementar DevOps de manera efectiva, existen diversas herramientas que facilitan la automatización, la colaboración y el monitoreo. Entre las herramientas más utilizadas en la implementación de DevOps se encuentran:

1.1.2.1. **Git y GitHub/GitLab**: Estas plataformas permiten la gestión y el control de versiones del código, facilitando la colaboración entre los desarrolladores. Git permite a los equipos trabajar en paralelo, mientras que GitHub y GitLab ofrecen entornos colaborativos donde se pueden revisar, probar y fusionar cambios de manera eficiente (Tiwari & Joseph, 2021).

1.1.2.2. **Jenkins**: Jenkins es una herramienta de integración continua ampliamente utilizada en DevOps. Su función principal es automatizar la integración y las pruebas del código, permitiendo detectar errores de manera temprana y agilizando el ciclo de desarrollo (Forsgren et al., 2018).

1.1.2.3. **Docker**: Docker permite la creación de contenedores, lo que facilita el despliegue y la portabilidad de las aplicaciones en diferentes entornos. Con Docker, los equipos de desarrollo y operaciones pueden asegurar que las aplicaciones funcionen de manera consistente, independientemente del entorno en el que se implementen (Kim et al., 2016).

1.1.2.4. **Kubernetes**: Kubernetes es una herramienta de orquestación de contenedores que permite gestionar aplicaciones en múltiples entornos de producción de manera escalable y eficiente. Kubernetes es particularmente útil en proyectos de ciencia de datos donde la escalabilidad y la eficiencia en el manejo de grandes volúmenes de datos son esenciales (Tiwari & Joseph, 2021).

DevOps representa un cambio fundamental en la forma en que las organizaciones desarrollan y operan el software, promoviendo una cultura de colaboración y mejora continua. La implementación de DevOps permite a las organizaciones no solo adaptarse a entornos competitivos, sino también ofrecer productos y servicios de alta calidad a un ritmo acelerado. Con herramientas y prácticas que promueven la automatización y la integración continua, DevOps facilita la adopción de metodologías ágiles en proyectos de ciencia de datos, habilitando el despliegue eficiente de modelos y aplicaciones basadas en datos.

1.2. Introducción a DataOps: Industrialización y Automatización en la Ciencia de Datos

DataOps, abreviatura de Data Operations, es una metodología orientada a la integración y automatización de los flujos de trabajo en ciencia de datos y análisis de datos. Inspirado en los principios de DevOps, DataOps se centra en optimizar cada paso del ciclo de vida de los datos, desde la recopilación y preparación hasta la implementación y monitoreo de modelos analíticos. Su objetivo es mejorar la colaboración entre los equipos de datos, acelerar los tiempos de respuesta en los proyectos de análisis y garantizar la calidad y consistencia de los datos para una toma de decisiones confiable y oportuna (Mishra & Sarawagi, 2019).

DataOps ha ganado popularidad debido a la creciente complejidad de los datos y la necesidad de respuestas rápidas en entornos empresariales competitivos. Las organizaciones que adoptan Data-

Ops pueden adaptarse mejor a los cambios y demandas del mercado, aprovechando sus datos de manera ágil y eficiente. DataOps también desempeña un papel clave en la ciencia de datos avanzada, facilitando la implementación de modelos de machine learning y análisis predictivo en entornos de producción (Burtch, 2018).

Enlace relacionado.

- ¿Qué es DataOps?
  Enlace: https://www.ibm.com/mx-es/topics/dataops
  En este artículo se aborda la metodología de DataOps y cómo optimiza la gestión de datos. Explora cómo DataOps mejora la calidad y consistencia de los datos en cada fase del ciclo de vida.

1.2.1. Principios Fundamentales de DataOps

DataOps se basa en varios principios que buscan unificar el flujo de trabajo de los datos y proporcionar una infraestructura confiable para su gestión (ver Figura 1).

Figura 1 Principios Fundamentales de DataOps

DataOps impulsa la automatización del ciclo de vida de los datos, desde la ingesta hasta la entrega de resultados, minimizando errores y tiempos de espera y permitiendo que los equipos se enfoquen en tareas de mayor valor agregado, evitando procesos repetitivos (Kim et al., 2016). También establece un monitoreo y control de calidad continuo de los datos, utilizando métricas para detectar errores en tiempo real y corregirlos antes de afectar los análisis y decisiones (Mishra y Sarawagi, 2019). DataOps adapta el modelo de integración y despliegue continuo (CI/CD) de DevOps a los flujos de datos, asegurando que los cambios en datos o modelos no afecten el rendimiento del sistema. Esto facilita pruebas y ajustes constantes en los pipelines, manteniendo la estabilidad del entorno de producción (Forsgren et al., 2018). Finalmente, fomenta una cultura de colaboración y responsabilidad compartida donde ingenieros, científicos de datos y analistas se comunican y trabajan juntos, asumiendo responsabilidad colectiva sobre la calidad y éxito del proyecto, lo cual es clave para la efectividad de DataOps (Erich et al., 2017).

1.2.2. Beneficios de DataOps en la Ciencia de Datos

La adopción de DataOps ofrece numerosos beneficios para los proyectos de ciencia de datos, especialmente en términos de velocidad, calidad y adaptabilidad. Uno de los beneficios principales de DataOps es la **agilidad en el desarrollo de proyectos analíticos**. Según Mishra y Sarawagi (2019),

DataOps permite que los equipos de datos implementen rápidamente modelos y soluciones, adaptándose a las demandas de un mercado cada vez más competitivo. Al automatizar los procesos de recolección y preparación de datos, DataOps minimiza el tiempo de espera entre las etapas del proyecto, acelerando el tiempo de respuesta y reduciendo los costos.

Otro beneficio importante es la **mejora en la calidad de los datos y en la confianza de los resultados analíticos**. Al incorporar controles de calidad y monitoreo continuo, DataOps garantiza que los datos sean precisos, completos y consistentes antes de ser utilizados en análisis o modelos de machine learning. Esto es especialmente importante en proyectos donde los errores en los datos pueden llevar a conclusiones equivocadas o decisiones erróneas. Kim et al. (2016) destacan que la implementación de DataOps ayuda a las organizaciones a obtener datos confiables y transparentes, lo cual es clave para una toma de decisiones informada.

Además, DataOps permite una **mayor flexibilidad y escalabilidad** en los proyectos de datos. La automatización y la integración continua facilitan la adaptación a nuevos requisitos o cambios en las fuentes de datos sin interrumpir el flujo de trabajo. En entornos donde los volúmenes de datos y los modelos analíticos crecen rápidamente, DataOps proporciona la infraestructura y los procesos necesarios para escalar de manera efectiva. Esto es especialmente relevante en ciencia de datos, donde los modelos y algoritmos deben ajustarse continuamente para mantenerse relevantes y efectivos (Burtch, 2018).

1.2.3. Herramientas y Prácticas en DataOps

La implementación de DataOps requiere de herramientas específicas que faciliten la automatización, la colaboración y el monitoreo de datos. Algunas de las herramientas más utilizadas en DataOps incluyen:

1.2.3.1. **Apache Kafka**: Kafka es una plataforma de procesamiento de eventos en tiempo real que permite la integración y el flujo de datos entre diferentes sistemas. En DataOps, Kafka facilita la transmisión continua de datos y permite que los equipos procesen grandes volúmenes de información en tiempo real (Tiwari & Joseph, 2021).

1.2.3.2. **Airflow**: Apache Airflow es una herramienta de orquestación de flujos de trabajo que permite programar, monitorear y gestionar pipelines de datos de manera eficiente. Con Airflow, los equipos pueden automatizar los procesos de extracción, transformación y carga (ETL) de datos, manteniendo la consistencia y calidad de estos (Mishra & Sarawagi, 2019).

1.2.3.3. **dbt (Data Build Tool)**: dbt es una herramienta que facilita la transformación de datos mediante código SQL. Permite a los equipos de datos definir y documentar los modelos de datos de manera reproducible, y validar su consistencia antes de cargarlos en los sistemas de análisis (Tiwari & Joseph, 2021).

1.2.3.4. **Great Expectations**: Esta herramienta de código abierto permite establecer y monitorear estándares de calidad en los datos, definiendo expectativas específicas para cada fuente o etapa del proceso de datos. Esto es fundamental en DataOps, ya que garantiza que los datos cumplan con criterios de calidad antes de ser utilizados en modelos o reportes (Forsgren et al., 2018).

1.2.4. Casos Prácticos de Implementación de DataOps

La aplicación de DataOps se puede observar en diversos sectores que requieren procesos analíticos rápidos y precisos. Por ejemplo, en la industria financiera, donde los cambios en el mercado y los patrones de consumo requieren de análisis en tiempo real, DataOps permite automatizar el flujo de datos y asegurar la consistencia de la información en cada punto del pipeline. Empresas de servicios financieros implementan DataOps para gestionar sus datos de manera eficiente, brindando información oportuna para la toma de decisiones estratégicas y operativas (Burtch, 2018).

Otro caso relevante es en el ámbito de la salud, donde los datos clínicos y de pacientes son críticos para el diagnóstico y tratamiento. Los hospitales y centros de investigación están adoptando DataOps para integrar grandes volúmenes de datos de múltiples fuentes (por ejemplo, registros de salud electrónicos, dispositivos médicos y estudios de imagen), permitiendo que los análisis y las predicciones clínicas se realicen con precisión y rapidez. La implementación de DataOps en estos entornos asegura que los datos sean fiables y estén actualizados, lo cual es vital para la salud de los pacientes (Tiwari & Joseph, 2021).

DataOps es una metodología esencial en la ciencia de datos moderna, ofreciendo un marco para optimizar la gestión y el análisis de datos en entornos complejos y de gran escala. A través de la automatización, la integración continua y la colaboración, DataOps permite a las organizaciones no solo mejorar la calidad y confiabilidad de sus datos, sino también adaptarse a las demandas de un mercado en constante cambio. Con herramientas que garantizan la consistencia y precisión de los datos, DataOps se posiciona como un enfoque indispensable en la industrialización de la ciencia de datos y en la implementación de modelos analíticos a gran escala.

1.2.5. Relación e Impacto de DevOps y DataOps en la Ciencia de Datos

La ciencia de datos ha evolucionado para convertirse en un pilar fundamental en la toma de decisiones estratégicas en las organizaciones. Sin embargo, la implementación efectiva de proyectos de ciencia de datos en entornos de producción enfrenta numerosos desafíos relacionados con la escalabilidad, la eficiencia y la colaboración entre equipos. Es en este contexto que metodologías como **DevOps** y **DataOps** han surgido como respuestas efectivas para mejorar los procesos de desarrollo y despliegue en proyectos de datos. Mientras que DevOps se centra en integrar los procesos de desarrollo y operaciones, DataOps extiende estos principios a la administración y procesamiento de datos, permitiendo a las organizaciones desplegar y gestionar sus soluciones analíticas de manera eficiente y confiable (Kim et al., 2016).

1.2.6. Sinergias entre DevOps y DataOps

DevOps y DataOps comparten los pilares de automatización, colaboración e integración continua, complementándose en ciencia de datos para gestionar proyectos precisos y eficientes. DevOps optimiza el ciclo de vida del software, mientras DataOps aplica estos principios a los datos, asegurando calidad y disponibilidad en el pipeline analítico (Mishra & Sarawagi, 2019). La implementación de pipelines CI/CD permite actualizar modelos de datos y analíticos de forma ágil y confiable, manteniendo la precisión en entornos de producción. Esto garantiza análisis y predicciones en tiempo real, alineando ambos enfoques para mejorar la eficiencia y confiabilidad de los proyectos de ciencia de datos (Forsgren et al., 2018).

1.2.7. Beneficios de DevOps y DataOps en la Ciencia de Datos

La combinación de DevOps y DataOps trae múltiples beneficios a los proyectos de ciencia de datos, entre los que se destacan la velocidad en el despliegue de modelos, la mejora en la calidad de los datos y la escalabilidad de las soluciones analíticas. Estos beneficios no solo optimizan el proceso de

desarrollo y operación, sino que también generan un valor significativo para las organizaciones, que pueden tomar decisiones basadas en datos de manera más ágil y confiable.

La adopción de DevOps y DataOps impulsa la agilidad y velocidad en el despliegue de modelos analíticos, permitiendo que los equipos implementen modelos de machine learning hasta un 50% más rápido, facilitando la adaptación a cambios de mercado Burtch (2018). En cuanto a la calidad de los datos, DataOps establece controles en cada fase del ciclo de vida, garantizando datos consistentes y libres de errores, lo que fortalece la precisión de los modelos y la toma de decisiones basada en datos Kim et al. (2016) . Además, estos enfoques permiten la escalabilidad y mantenimiento de soluciones analíticas, con herramientas como Kubernetes y Docker, que facilitan la implementación reproducible en diferentes entornos (Tiwari & Joseph, 2021).

1.2.8. Desafíos en la Implementación de DevOps y DataOps en Ciencia de Datos

Aunque DevOps y DataOps ofrecen numerosos beneficios, su implementación en ciencia de datos también presenta ciertos desafíos que las organizaciones deben considerar. Algunos de los principales desafíos incluyen la complejidad en la gestión de datos, la cultura organizacional y la necesidad de herramientas avanzadas para monitorear y gestionar los pipelines de datos.

Los proyectos de ciencia de datos enfrentan varios desafíos clave. La gestión de datos es compleja debido a los grandes volúmenes y diversidad de fuentes, lo que requiere pipelines especializados y herramientas robustas para integrar y gestionar datos eficientemente Mishra y Sarawagi (2019) . Además, la implementación de DevOps y DataOps exige un cambio cultural y colaboración entre científicos de datos, ingenieros y desarrolladores, con una cultura organizacional que fomente la comunicación y responsabilidad compartida (Erich et al., 2017). Finalmente, la necesidad de herramientas especializadas como Apache Kafka, Apache Airflow y Kubernetes es esencial para automatizar y monitorear flujos de trabajo en entornos complejos y escalables (Forsgren et al., 2018).

A continuación, en la Figura 2, podemos ver un resumen de la Relación e Impacto, los Beneficios y Desafíos, y las Sinergias entre DevOps y DataOps en la Ciencia de Datos.

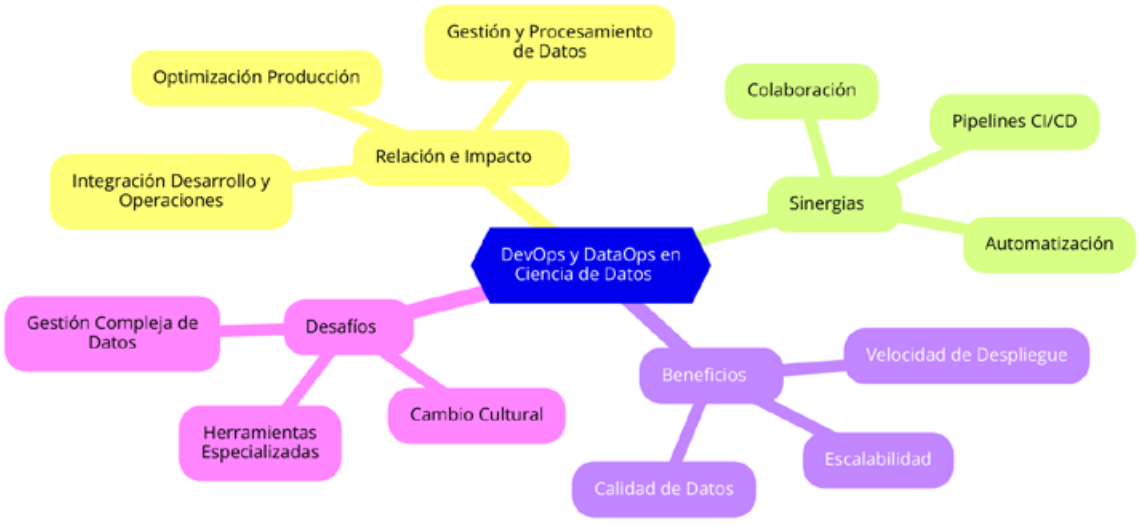

Figura 2 DevOps y DataOps en la Ciencia de Datos

1.2.9. Casos Prácticos de DevOps y DataOps en Ciencia de Datos

La implementación de DevOps y DataOps en ciencia de datos se puede observar en sectores como la

banca, el comercio electrónico y la salud, donde el análisis y la precisión en los datos son fundamentales para la toma de decisiones.

En el sector financiero, DevOps y DataOps permiten analizar grandes volúmenes de datos en tiempo real, facilitando la detección temprana de fraudes y personalizando servicios al cliente mediante datos confiables y actualizados (Burtch, 2018). En comercio electrónico, ambas metodologías mejoran la experiencia del usuario y optimizan el marketing mediante el análisis de preferencias de compra, integrando fuentes de datos como historial de navegación y demografía para generar recomendaciones precisas (Tiwari & Joseph, 2021). En salud y ciencias de la vida, DataOps facilita la integración de datos de salud electrónica, dispositivos y estudios, permitiendo diagnósticos más precisos a través de modelos predictivos actualizados continuamente (Kim et al., 2016).

La combinación de DevOps y DataOps en la ciencia de datos representa una evolución significativa en la forma en que las organizaciones gestionan sus proyectos analíticos. Estos enfoques no solo permiten mejorar la velocidad y calidad en el despliegue de modelos, sino que también aseguran la escalabilidad y confiabilidad de las soluciones analíticas. La implementación de DevOps y DataOps requiere de una infraestructura robusta y una cultura organizacional colaborativa, pero los beneficios en términos de eficiencia, adaptabilidad y precisión en los datos son notables. Al adoptar estos enfoques, las organizaciones pueden optimizar sus procesos de ciencia de datos y posicionarse de manera competitiva en un mercado impulsado por la información.

Referencias citadas en la Clase 6.

Burtch, B. (2018). *DataOps: A New Methodology for Data Management*. Database Trends and Applications, 32(2), 45-49.

Erich, F., Amrit, C., & Daneva, M. (2017). *A Qualitative Study of DevOps Usage in Practice*. Journal of Software: Evolution and Process, 29(6), e1885.

Forsgren, N., Humble, J., Kim, G., & Kersten, M. (2018). *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. Portland: IT Revolution Press.

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations*. Portland: IT Revolution Press.

Mishra, S., & Sarawagi, M. (2019). *DataOps: Industrialization of Data Management*. O'Reilly Media.

Tiwari, A., & Joseph, A. (2021). *Practical DataOps: Delivering Agile Data Science at Scale*. Packt Publishing.

Definición de los términos citados en la Clase 6.

1. **Integración Continua y Entrega Continua (CI/CD)**: Una práctica de desarrollo que permite la integración frecuente de código y su despliegue rápido en producción, facilitando actualizaciones sin interrupciones.
2. **Pipeline de Datos**: Serie de procesos automáticos para la ingesta, transformación y análisis

de datos, desde la fuente hasta el usuario final o el sistema de análisis.

Profundización Clase 6.

1. Casos de Automatización en Ciencia de Datos

La automatización es un pilar fundamental tanto en DevOps como en DataOps, ya que permite a los equipos de datos y software concentrarse en actividades de mayor valor. En el contexto de ciencia de datos, la automatización incluye la preparación de datos, el entrenamiento de modelos y el despliegue en producción. Un ejemplo común es la automatización de pipelines de datos mediante herramientas como Apache Airflow. Esta herramienta permite programar procesos de extracción, transformación y carga de datos (ETL) de forma automática, asegurando que los datos estén listos para su análisis en cada actualización, lo cual es esencial para el aprendizaje automático.

Configura y ejecuta un pipeline de datos automatizado utilizando Apache Airflow para comprender cómo esta herramienta optimiza procesos de extracción, transformación y carga (ETL) en proyectos de ciencia de datos.

Ejercicio:

- **Instalación y Configuración Inicial:**
  - Instala Apache Airflow en tu máquina local o en un entorno virtual (puedes usar Docker para simplificar el proceso).
  - Configura un entorno de trabajo con Airflow y asegúrate de que el servidor web de Airflow esté activo.
- **Definición del Pipeline:** Diseña un pipeline simple que incluya las siguientes tareas:
  - **Extracción:** Lee un archivo CSV desde un repositorio local o remoto.
  - **Transformación:** Filtra los datos para incluir solo las columnas necesarias y realiza una limpieza básica.
  - **Carga:** Guarda los datos transformados en una base de datos SQLite o en un archivo nuevo.
- **Implementación del DAG:**
  - Define el DAG (Directed Acyclic Graph) en un archivo .py.
  - Configura las dependencias entre las tareas para que se ejecuten en el orden correcto.
- **Prueba del Pipeline:**
  - Ejecuta el pipeline desde el servidor web de Airflow.
  - Verifica los resultados en la base de datos o archivo de salida.

Entrega:

- Capturas de pantalla del pipeline ejecutándose en el servidor web de Airflow.
- Un breve informe (máximo 1 página) explicando:

- La lógica detrás del diseño del pipeline.
- Los desafíos encontrados y cómo los solucionaste.
- Cómo esta automatización podría beneficiar a un equipo de ciencia de datos.

2. Integración Continua y Despliegue Continuo (CI/CD) en Ciencia de Datos

Aprender a configurar un pipeline de CI/CD utilizando Jenkins para garantizar la integración, pruebas y despliegue continuo de un modelo de machine learning en un entorno controlado. Implementar un pipeline de CI/CD asegura que los cambios en el modelo y en los datos sean probados antes de su despliegue, evitando interrupciones en el sistema de producción.

Tareas:

- **Preparación del Entorno:**
  - Instala Jenkins en tu máquina local o en un servidor en la nube.
  - Configura un repositorio en GitHub o GitLab que contenga:
    - El código de entrenamiento del modelo (puedes usar un ejemplo básico en Python, como un modelo de clasificación con scikit-learn).
    - Un conjunto de datos de prueba.
    - Un script de despliegue simulado que almacene el modelo entrenado en un directorio.
- **Diseño del Pipeline CI/CD:**

  Configura un pipeline en Jenkins que incluya las siguientes etapas:

  1. **Integración:** Verifica que el código subido al repositorio cumple con los estándares definidos mediante una herramienta como pylint.
  2. **Pruebas:** Ejecuta un script que entrene el modelo con datos de prueba y valide su rendimiento (por ejemplo, verificando una métrica como precisión o F1-score).
  3. **Despliegue:** Si el modelo cumple con los requisitos de calidad, guárdalo en un directorio simulado de producción.

- **Implementación del Pipeline:**
  - Utiliza el archivo de configuración de Jenkins (Jenkinsfile) para definir el pipeline.
  - Asegúrate de incluir pasos para notificar fallas o éxitos (puedes usar correo electrónico o Slack).
- **Prueba del Pipeline:**
  - Realiza un cambio en el código del modelo (por ejemplo, modifica un hiperparámetro) y observa cómo Jenkins ejecuta automáticamente el pipeline.

- Revisa los logs de Jenkins para asegurarte de que todas las etapas se ejecuten correctamente.

Entrega:

- Capturas de pantalla de las ejecuciones del pipeline en Jenkins.
- Un breve informe (máximo 1 página) que explique:
  - La lógica detrás del diseño del pipeline.
  - Los desafíos encontrados y cómo los solucionaste.
  - Los beneficios de usar CI/CD en proyectos de ciencia de datos.

PUCE Virtual

La excelencia no se improvisa

síguenos

MDASC - 12 - 241