UNIVERSIDAD PRIVADA SAN JUAN BAUTISTA

FACULTAD DE INGENIERÍAS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

INTEGRACION DE FRAMEWORK DEVSECOPS EN PROYECTOS MLOPS

TRABAJO DE SUFICIENCIA PROFESIONAL

PRESENTADA POR BACHILLER

ROJAS DENEGRI JOSE FERNANDO

PARA OPTAR EL TÍTULO PROFESIONAL DE

INGENIERO DE COMPUTACIÓN Y SISTEMAS

LIMA - PERÚ

2025

ASESOR Y AUTOR

ASESOR: JOSE ANTONIO OGOSI AUQUI

CODIGO ORCID: 0000-0002-4708-610X

AUTOR: ROJAS DENEGRI JOSE FERNANDO

CODIGO ORCID: 0009-0005-9659-9090

AGRADECIMIENTO

Agradezco a mi familia por el apoyo en todo este camino, y a mi novia por motivarme a seguir adelante.

DEDICATORIA

Dedicado a familia, novia que fueron mis pilares para seguir adelante, la confianza que depositaron y depositan para continuar en el camino del éxito.

RESUMEN

Este trabajo nace con la intención de dar un paso adelante en cómo se manejan los proyectos de machine learning dentro de entornos reales. La idea principal no es solo aplicar DevSecOps, sino construir una forma de trabajo que ayude a cuidar la seguridad, agilizar procesos y mejorar la calidad desde el primer momento en que se empieza a desarrollar un modelo. Todo esto se pensó específicamente para entornos de MLOps, donde cada cambio debe quedar registrado y ser fácil de rastrear, algo clave cuando se habla de modelos que impactan decisiones reales.

Para lograrlo, se combinaron distintos enfoques: DevOps, SecOps y MLOps. Se diseñaron flujos de trabajo en GitHub y se centralizó la gestión de secretos con herramientas como Hashicorp Vault. También se integraron análisis de vulnerabilidades usando SonarQube, Fortify y Orca Security, según lo que cada proyecto necesitaba. A esto se sumaron automatizaciones que se encargan por sí solas de ejecutar pruebas, desplegar artefactos y monitorear su comportamiento, incluyendo validaciones de código y escaneos permanentes de imágenes.

Durante las pruebas iniciales y la marcha blanca se notaron mejoras claras: los despliegues se volvieron más rápidos, el riesgo de filtrar credenciales bajó notablemente y los modelos comenzaron a operar con mayor confianza y estabilidad. Estos avances demostraron que el enfoque no solo funcionaba, sino que podía adaptarse a distintos escenarios sin perder control ni seguridad.

En resumen, el marco planteado abre espacio para que la seguridad no sea un añadido tardío, sino parte del proceso desde el inicio. Además, facilita que los proyectos puedan cumplir con normas y requisitos técnicos sin perder agilidad, algo clave cuando se busca escalar soluciones de análisis avanzado o inteligencia artificial.

Palabras clave: DevSecOps, MLOps, automatización, seguridad, CI/CD, infraestructura como código.

ABSTRACT

This work was conceived with the goal of taking a step forward in how machine learning projects are managed in real-world environments. The main idea goes beyond simply applying DevSecOps—it aims to establish a way of working that protects security, speeds up processes, and improves quality from the very first stage of model development. Everything was designed with MLOps environments in mind, where every change must be recorded and easy to trace—especially when dealing with models that influence real decisions.

To achieve this, different approaches were combined: DevOps, SecOps, and MLOps. Workflows were designed in GitHub, and secret management was centralized using tools like Hashicorp Vault. Vulnerability analysis was also integrated through tools such as SonarQube, Fortify, and Orca Security, depending on each project's needs. In addition, automations were implemented to run tests, deploy artifacts, and monitor their behavior automatically, including code validations and continuous image scanning.

During the initial testing and pilot phase, clear improvements were observed: deployments became faster, the risk of credential exposure significantly decreased, and the models began operating with greater reliability and stability. These results showed that the approach not only worked but was also flexible enough to adapt to different scenarios without losing control or security.

In summary, the proposed framework makes it possible for security to be integrated from the start, rather than added at the end. It also enables projects to meet technical and regulatory requirements without sacrificing agility—something essential when scaling advanced analytics or artificial intelligence solutions.

Keywords: DevSecOps, MLOps, cloud automation, cybersecurity, CI/CD pipelines, infrastructure as code.

INTRODUCCIÓN

Hoy en día, desarrollar software sin una forma clara de organizar el trabajo puede generar muchos problemas. No basta con escribir código y ya; se necesita un camino que guíe todo el proceso para que el producto sea seguro, funcione bien y esté listo cuando se necesite. A pesar de eso, todavía existen empresas que siguen métodos lentos y muy manuales, lo que provoca retrasos y decisiones tomadas “a ciegas”, sin información actualizada. En este proyecto se analizó un caso real que evidencia justamente eso: que los enfoques tradicionales pueden afectar directamente las decisiones estratégicas e incluso provocar pérdidas económicas por no contar con datos en tiempo real.

Durante el desarrollo del trabajo, se exploró cómo la incorporación de un enfoque DevSecOps desde las primeras etapas del ciclo de vida trae mejoras evidentes. No solo ayuda a mantener la calidad del producto, sino que también refuerza la seguridad y facilita la colaboración entre los equipos. Con un marco así, los tiempos de entrega se reducen y se detectan fallos o riesgos mucho antes, evitando retrabajos y cuellos de botella.

Más que una serie de pasos, DevSecOps se entiende como una forma de trabajar: una combinación de buenas prácticas, herramientas y decisiones que permiten desplegar soluciones seguras y confiables, alineadas con los objetivos del negocio. En otras palabras, sirve para que la tecnología y la estrategia vayan en la misma dirección.

ÍNDICE

LISTA DE TABLAS

LISTA DE FIGURAS

INFORME ANTIPLAGIO

ROJAS DENEGRI JOSE FERNANDO

01 TRAB. SUF. PROF. - ROJAS DENEGRI JOSE FERNANDO

TRAB. SUF. POR TALLER 2025-1

Detalles del documento

Identificador de la entrega
trn:oid:::3117:534123302

Fecha de entrega
29 nov 2025, 17:13 GMT-5

Fecha de descarga
12 dic 2025, 15:51 GMT-5

Nombre del archivo
01 TRAB. SUF. PROF. - ROJAS DENEGRI JOSE FERNANDO.docx

Tamaño del archivo
8.6 MB

47 páginas

6341 palabras

37.159 caracteres

turnitin Página 1 de 50 - Portada Identificador de la entrega trn:oid:::3117:534123302

1% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Coincidencias menores (menos de 15 palabras)

Fuentes principales

1%	Fuentes de Internet
0%	Publicaciones
1%	Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alertas de integridad para revisión

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

turnitin Página 2 de 50 - Descripción general de integridad Identificador de la entrega trn:oid:::3117:534123302

*% detectado como IA

La detección de IA incluye la posibilidad de que haya falsos positivos. Aunque cierto texto en esta entrega se generó probablemente con IA, los puntajes inferiores al umbral del 20 % no aparecen porque tienen una mayor probabilidad de falsos positivos.

Precaución: Se necesita revisión.

Es esencial comprender los límites de la detección de IA antes de tomar decisiones acerca del trabajo del estudiante. Te alentamos a obtener más información acerca de las funciones de detección de IA de Turnitin antes de usar la herramienta.

Preguntas frecuentes

¿Cómo debería interpretar los falsos positivos y el porcentaje de escritura con IA de Turnitin?
El porcentaje que se muestra en el reporte de escritura con IA es la cantidad del texto calificado en la entrega que el modelo de detección de escritura con IA de Turnitin determina se generó probablemente con IA desde un modelo de lenguaje de gran tamaño.

Los falsos positivos (que marcan incorrectamente alertas de texto escrito por humanos como generado con IA) son una posibilidad en los modelos de IA.

Los puntajes de detección de IA inferiores al 20 %, que no aparecen en reportes nuevos, tienen una mayor probabilidad de ser falsos positivos. Para reducir la probabilidad de malinterpretación, no se atribuye ningún puntaje o resaltado y se indican con un asterisco en el reporte (*%).

El porcentaje de escritura con IA no debe ser el único fundamento para determinar si ha ocurrido una mala conducta. El revisor/instructor debería usar el porcentaje como un medio para iniciar una conversación formativa con sus estudiantes o usarlo para examinar el ejercicio entregado según las políticas de la escuela.

¿Qué significa 'texto calificado'?
Nuestro modelo sólo procesa texto calificado en la forma de escritura de formato largo. La escritura de formato largo se refiere a los enunciados individuales en párrafos que constituyen una parte más grande del trabajo escrito, como un ensayo, una disertación, un artículo, etc. El texto calificado que se ha determinado que se generó probablemente con IA se resaltará en color cian en la entrega.

El texto no calificado, como viñetas, bibliografías comentadas, etc., no se procesará y puede crear disparidad entre los puntos destacados de la entrega y el porcentaje mostrado.

turnitin Página 2 de 49 - Descripción general de la escritura con IA Identificador de la entrega trn:oid:::3117:534123302

CAPÍTULO I
DESCRIPCIÓN DEL CENTRO LABORAL Y FUNCIONES

Kyndryl Perú S.A.C. forma parte de una compañía internacional que nació con un objetivo claro: ayudar a las empresas a adaptarse al mundo digital y a modernizar sus operaciones a través de la tecnología. En el contexto peruano, su presencia se ha vuelto relevante porque muchas organizaciones necesitan manejar sistemas complejos mientras avanzan hacia soluciones en la nube, automatización, ciberseguridad y un manejo más inteligente de la información.

Su llegada al país no fue un hecho aislado. Después de separarse de IBM en 2021, la empresa comenzó a expandirse y encontró en Perú un terreno con alta demanda de servicios tecnológicos de calidad. Hoy en día, colabora con sectores como la banca, telecomunicaciones, comercio y servicios públicos, acompañando a estas industrias en su transición hacia procesos digitales más sólidos y eficientes.

Pero Kyndryl Perú no se limita únicamente a administrar infraestructura. También busca fomentar la innovación y ofrecer a las empresas la agilidad que necesitan para mantenerse competitivas. Trabaja de la mano con compañías líderes del país, con el propósito de prepararlas frente a los desafíos de un entorno que cambia constantemente y donde la tecnología se ha convertido en un factor clave para crecer.

Misión Kyndryl Perú S.A.C.

La misión de Kyndryl en Perú se centra en acompañar a las empresas en su camino hacia la transformación tecnológica. No se trata solo de proveer infraestructura o mover todo a la nube, sino de ofrecer soluciones completas que ayuden a trabajar mejor, reaccionar más rápido y mantener los sistemas protegidos frente a cualquier imprevisto. La idea es que las organizaciones puedan adaptarse al entorno digital sin perder estabilidad, mejorando sus procesos clave y apostando por un crecimiento sostenible y sólido. En pocas palabras, Kyndryl busca impulsar la tecnología como base para la eficiencia y la excelencia empresarial.

Visión Kyndryl Perú S.A.C.

Kyndryl aspira a convertirse en el aliado tecnológico más confiable del país, especialmente en los momentos en que las empresas deciden dar el salto a lo digital. La meta es acompañar a las organizaciones peruanas en la construcción de una base

tecnológica sólida, que no solo permita modernizar sus operaciones, sino también impulsarlas a competir a nivel digital. Se busca liderar este camino a través de soluciones innovadoras, eficiencia en la operación diaria y un compromiso real con la sostenibilidad. En otras palabras, ser el motor que ayude al país a avanzar tecnológicamente con creatividad y visión de futuro.

1.1 DESCRIPCIÓN DEL CENTRO LABORAL

1.1.1 RAZÓN SOCIAL

KYNDRYL PERÚ S.A.C.

Ubicación: Av. Javier Prado Este 6230, La Molina 15012

Website: https://www.kyndryl.com/pe/es

1.1.2 SECTOR AL QUE PERTENECE

Servicios de Infraestructura de Tecnologías de la Información (TI) y Servicios Gestionados de TI

Kyndryl Perú S.A.C. se dedica a brindar servicios tecnológicos enfocados en mantener en buen estado los sistemas que soportan operaciones críticas. Su trabajo no solo se queda en administrar infraestructura, sino que también busca que todo funcione con altos niveles de seguridad y eficiencia. Para ello, emplean tecnologías que permiten crear entornos más flexibles y conectados, como Kubernetes, Docker, OpenShift o VMware, los cuales facilitan el uso de contenedores y la comunicación entre diferentes plataformas.

Además, la empresa impulsa soluciones en la nube, tanto híbridas como multinube, trabajando con proveedores como AWS, Azure o Google Cloud. Esto les permite adaptarse a lo que cada organización necesita y escalar sus recursos sin depender de una sola plataforma. Otro aspecto clave es la automatización: usan herramientas como Terraform, Jenkins, Ansible o GitHub Actions para reducir pasos manuales y acelerar las implementaciones, algo esencial para evitar retrasos y errores humanos.

A todo esto se suma la integración de sistemas de monitoreo como Datadog, Grafana o Dynatrace, que sirven para detectar problemas antes de que afecten la operación. Y para proteger toda esta infraestructura, aplican estrategias de ciberseguridad basadas en el enfoque Zero Trust, lo que asegura que cada acceso sea verificado y que la información esté protegida en todo momento. En conjunto, se trata de un modelo que busca eficiencia, seguridad y sostenibilidad operativa.

1.1.3 ESTRUCTURA ORGANIZACIONAL

Figura 1

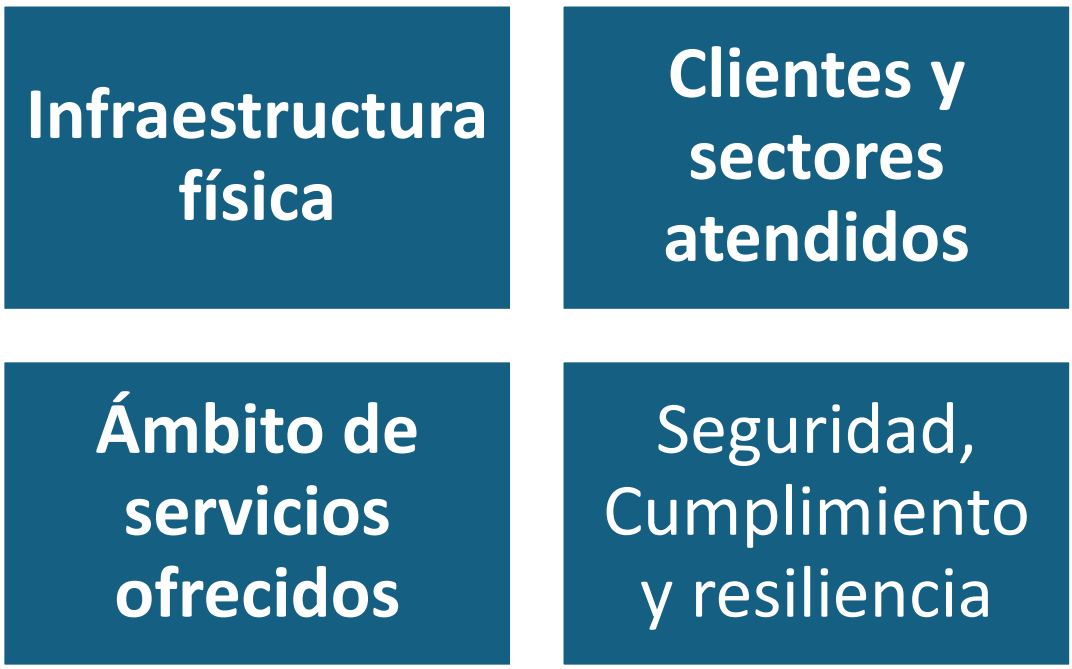


Tabla 1

Dimensión	Proyectos Representativos
Infraestructura física	Operación de data center en Lima con servicios de operación, respaldo y recuperación ante desastres. Estos proyectos de consolidación de infraestructura crítica direccionado a entidades financieras.
Clientes y sectores atendidos	Atiende banca, telecomunicaciones, retail y seguros. Un gran reflejo de esto es la

	alianza con BCP para modernizar infraestructura tecnológica con Microsoft.
Ámbito de servicios ofrecidos	Cobertura de todo el ciclo de vida tecnológico (migración a la nube, gestión de datos, ciberseguridad, espacios de trabajo digitales). Un proyecto de implementación de plataformas multicloud en retail para soportar campañas de ventas masivas.
Seguridad, Cumplimiento y resiliencia	Cumplimiento de normas internacionales y locales (ISO 27001, PCI-DSS, SBS). Orquestación de planes de continuidad y recuperación ante desastres para el sector financiero, además de proyectos de resiliencia cibernética en telecomunicaciones.

1.1.4 ÁREA DE DESEMPEÑO

Estructura Regional de Dirección Perú

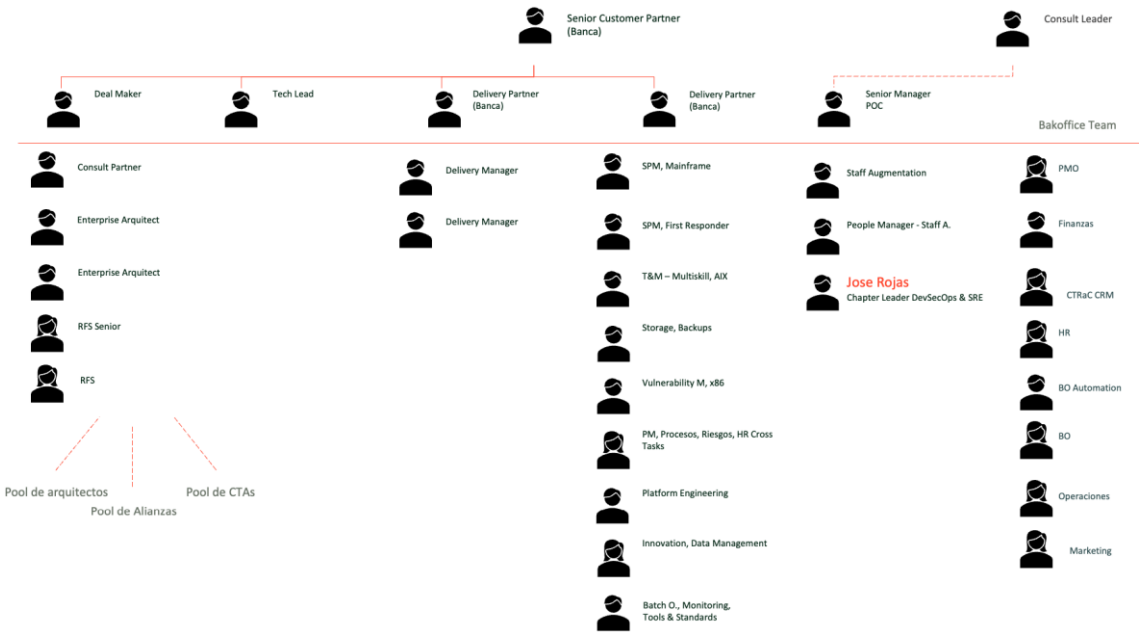
La estructura de la entidad se compone de diversos roles estratégicos y de apoyo, tales como Socios Senior de Clientes (por industria), Líder de Consultoría, Líderes de CTAs y CTOs, Alianzas, y Comunicación y Relaciones Públicas, junto con áreas de apoyo esenciales: director de Operaciones, Recursos Humanos y Finanzas. También se incluyen funciones especializadas como la Integración de Entregas y el Líder de CU Multi-Industria, que facilitan el respaldo a las operaciones y satisfacer las necesidades de variados grupos de clientes.

Figura 2

Estructura de Perú

El diagrama de organización muestra la manera en que se organizan los diversos niveles de responsabilidad: desde la gerencia, los socios clave, hasta los grupos de implementación, diseño, automatización, protección, información, programas y asistencia.

Figura 3



Servicios del Área

Kyndryl Perú S. A. C. proporciona una gama completa de soluciones tecnológicas destinadas a satisfacer las demandas actuales de transformación digital de las empresas. Estas soluciones no solo tratan aspectos tecnológicos, sino que también facilitan la mejora de la eficiencia operativa, la seguridad, la sostenibilidad y la toma de decisiones estratégicas en un mercado empresarial muy competitivo. La propuesta se organiza en diversas áreas especializadas que incluyen desde la migración a la nube hasta la inteligencia artificial, la automatización, la resiliencia operativa y la administración del entorno digital laboral.

1. Servicios en nube (Cloud Services)
    - Migración al cloud.
    - Modernización de la nube.
    - Infraestructura de nube privada como servicio (IaaS).
    - Gestión multicloud.
2. Aplicaciones, datos e inteligencia artificial
    - Modernización de aplicaciones empresariales.
    - Gestión de datos e inteligencia artificial / analítica.
    - Optimización del rendimiento de ERP (como SAP, Oracle) en entornos de nube.
3. Modernización de infraestructura crítica ("Core Enterprise / zCloud / Mainframe")
    - Servicios de mainframe y modernización de mainframe.
    - Infraestructura híbrida y privada, optimización de componentes antiguos.
4. Redes y Edge
    - Servicios de redes definidas por software (SDN), SD-WAN.
    - Tecnologías Edge, conectividad perimetral.
5. Seguridad y resiliencia (Cybersecurity & Resilience)
    - Protección de datos, respaldo (backup) como servicio.
    - Recuperación ante desastres (Disaster Recovery).
    - Orquestación de resiliencia de TI.
6. Espacio de trabajo digital
    - Gestión de dispositivos.
    - Virtualización de escritorios.

- Servicios de colaboración y soporte tecnológico para los puestos de trabajo.

7. Consultoría y servicios gestionados
    - Consultoría estratégica en transformación digital.
    - Servicios gestionados (infraestructura, redes, seguridad, aplicaciones).
8. Kyndryl Bridge
    - Plataforma de integración abierta para unir negocio digital y tecnología, facilitar la orquestación, acelerar innovaciones.

Fuente: https://www.kyndryl.com/pe/es/services

Cargo: Chapter Leader DevSecOps & SRE

Como CL DevSecOps & SRE, estamos en búsqueda de la eficiencia y obtención de resultados que permitan dar mayor beneficio a la organización:

a) Implementar flujos de CI/CD seguros donde reducen desde un 40% a 50% en los tiempos de despliegue y minimizan los errores humanos en entornos productivos.
b) Incluir herramientas para Infraestructura como código, esto permite estandarizar los despliegues de infraestructura donde se logran tiempos record en el despliegue o aprovisionamiento y en los esquemas de recuperación ante desastres.
c) Incorporar herramientas de monitoreo, este enfoque de Observabilidad permiten anticiparte a caídas de servicio como aplicación o a nivel de infraestructura, otorgando información correcta al equipo de operaciones y otorgando un análisis causa raíz más precisa.

1.2 DESCRIPCIÓN DE LAS FUNCIONES DESEMPEÑADAS Y SU VINCULACIÓN CON CAMPOS TEMÁTICOS DE LA CARRERA PROFESIONAL

Como Chapter Leader DevSecOps & SRE en Kyndryl Perú S.A.C., se cumple una función estratégica dentro de la organización, al ser el responsable de liderar y fomentar las prácticas de desarrollo seguro, confiable y automatizado de infraestructura tecnológica acorde a los altos estándares de calidad que exige la industria.

Los Principales roles son los siguiente:

- Liderazgo técnico y cultural
    a. Guiar a los equipos en la adopción de las practicas DevSecOps, permitiendo integrar la seguridad en todas las fases de desarrollo de software.
    b. Promover una cultura de SRE, orientada a la disponibilidad, escalabilidad y resiliencia.
- Gestión y coordinación de Chapter's
    a. Definir buenas prácticas y lineamientos donde permita ser guía importante para toda la organización.
    b. Fomentar comunidad "Sharing" donde el compartir conocimiento sea uno de los principales pilares.
- Gobernanza y cumplimiento
    a. Guiar a los equipos de desarrollo en obtener contemplar altos estándares de calidad de software involucrando la seguridad como un factor a implementar en etapa temprana de inicio de proyecto.
    b. Definir la integración de herramienta que permitan controlar la calidad, seguridad, control y monitorio de aplicaciones, incluyendo los esquemas de despliegue como CI/CD.
- Innovación y automatización
    a. Impulsar la adopción de nuevas tecnologías de Observabilidad, IaC, automatización y despliegues, incluyendo herramientas desde la CNCF (Cloud Native Computing Foundation).
    b. Evaluar eficiencia involucrando FinOps como estrategia en cada proyecto tecnológico.
- Relación con clientes y negocio
    a. Seguimiento a los equipos en la implementación de soluciones seguras y confiables.
    b. Asegurar la continuidad operativa y resiliencia digital, incluyendo la adopción de modelos operativos.

CAPÍTULO II
SITUACIONES PROBLEMÁTICAS Y CONTRIBUCIONES

Cuando comenzó el proceso de migración desde los servidores locales hacia la nube, quedaron en evidencia varias fallas que antes pasaban desapercibidas. Los lanzamientos de software tomaban demasiado tiempo, la protección de los entornos no era uniforme y el funcionamiento diario dependía mucho del trabajo manual. Todo esto afectaba la productividad y mostraba que se necesitaba un cambio urgente en la forma de trabajar. Era claro que hacía falta una estrategia que ordenara los procesos, redujera errores humanos y garantizara seguridad desde el principio hasta el final de cada desarrollo.

Frente a este panorama, se planteó adoptar un enfoque DevSecOps. La idea fue unir tres pilares —desarrollo, operaciones y seguridad— dentro de un mismo modelo, con el objetivo de automatizar la mayor parte del ciclo de vida del software. Con ello, se buscó acelerar los despliegues, incorporar controles de seguridad constantes y lograr que cada entrega sea más confiable que la anterior. Gracias a esta integración, fue posible reducir riesgos, mejorar la resistencia operativa y evitar fallos que antes eran muy comunes.

Más que una solución técnica, el uso de DevSecOps permitió que los equipos trabajaran de forma más coordinada. La seguridad dejó de ser una etapa final y empezó a considerarse desde la planificación, lo que transformó por completo la manera de construir y desplegar soluciones digitales.

2.1. DESCRIPCIÓN DE UNA SITUACIÓN PROBLEMÁTICA O CASO CLÍNICO QUE SE LE HUBIESE PRESENTADO EN EL AÑO QUE DESEMPEÑO SUS FUNCIONES

Despliegues lentos de modelos de machine learning

- Los modelos solo se publicaban cada seis meses, lo que impedía a otras áreas trabajar con datos actualizados.
- El procesamiento se hacía en infraestructura local, lo que limitaba la capacidad de cómputo y ralentizaba las operaciones.

Falta de un marco claro para el ciclo de desarrollo

- No había un modelo DevSecOps que guiara la creación y publicación de los modelos, lo que dejaba vacíos técnicos en fases clave del desarrollo.
- Al no evaluar desde el inicio, las aplicaciones analíticas tenían una respuesta lenta y poco eficiente.

Procesamiento en tiempo real insuficiente

- Trabajar con grandes volúmenes de información en tiempo real generaba demoras por falta de orquestación entre los componentes.
- Los modelos no se consumían en tiempo real, solo se usaban después del despliegue en producción.

Infraestructura sin automatización

- El aprovisionamiento de recursos y las pruebas se hacían manualmente, lo que alargaba los tiempos y no siempre garantizaba viabilidad técnica.
- No existían herramientas que permitieran una creación rápida y ordenada de infraestructura según las necesidades del proyecto.

2.2.CONTRIBUCIÓN EN LA SOLUCIÓN DE SITUACIÓN PROBLEMÁTICA

Con el fin de abordar las dificultades detectadas durante la fase de diagnóstico del proyecto, se elaboró e instauró una serie de medidas técnicas y metodológicas fundamentadas en la incorporación de un marco DevSecOps aplicado a contextos MLOps. Esta estrategia facilitó la fusión de la automatización, la seguridad y la efectividad operativa en cada etapa del ciclo de vida de los modelos de aprendizaje automático, mejorando los tiempos, los recursos y la calidad de las entregas.

Optimización de despliegues de modelos de machine learning

Para acortar los tiempos de implementación y eliminar la necesidad de procedimientos manuales:

- Se pusieron en marcha flujos de trabajo CI/CD mediante GitHub Actions, permitiendo automatizar las fases de creación, pruebas, empaquetado y lanzamiento de modelos.
- Se trasladó el procesamiento de infraestructura local a servicios en la nube, incrementando la potencia de cómputo y posibilitando el procesamiento distribuido a demanda.
- Se instituyó un modelo de entrega continua, logrando disminuir el intervalo de despliegue de seis meses a ciclos iterativos más rápidos y previsibles.

Incorporación de un ciclo de vida seguro y gobernado para desarrollo

Para abordar la carencia de evaluación previa y la falta de un esquema DevSecOps:

- Se creó un proceso de desarrollo uniforme fundamentado en las mejores prácticas de DevSecOps y MLOps.
- Se incorporaron fases de revisión de calidad y seguridad desde las etapas iniciales del desarrollo, abarcando la revisión de código, las pruebas unitarias y la validación de dependencias.
- Se implementó un escaneo de seguridad automatizado en cada pipeline utilizando herramientas como SonarQube, Fortify y Orca Security para detectar vulnerabilidades tecnológicas antes de la implementación.

Mejora del procesamiento de modelos en tiempo real

Para abordar la carencia de orquestación y el procesamiento fuera de línea:

- Se activaron flujos de trabajo de inferencia y reentrenamiento continuo, lo que facilitó el uso de modelos de forma planificada y a demanda.

- Se implementaron microservicios encapsulados en contenedores Docker organizados con Kubernetes para permitir la inferencia escalable en tiempo casi real.

- Se añadió supervisión de la latencia y el rendimiento del modelo en producción para ajustar de manera automática la capacidad de procesamiento.

Automatización del aprovisionamiento de infraestructura

Para abordar la escasez de disponibilidad y el aprovisionamiento manual:

- Se llevó a cabo la implementación de Infraestructura como Código (IaC) utilizando Terraform y ARM Templates, lo que posibilita la creación de entornos que se pueden reproducir.

- Se automatizó el aprovisionamiento de recursos para pipelines de datos, almacenamiento, computación y redes a través de integraciones con CI/CD.

- Se creó un catálogo de infraestructura reutilizable basado en plantillas validadas, aumentando la eficiencia y disminuyendo los tiempos de implementación en hasta un 60%.

Contribución en la Solución

Diseño del marco DevSecOps para MLOps

- Lideré el diseño de la arquitectura del framework DevSecOps integrando herramientas de automatización, seguridad y orquestacion.
- Definí el ciclo de vida de desarrollo para modelos de machine learning alineado a buenas prácticas conforme a la industria.
- Estandaricé flujos de trabajo para despliegues seguros y resilientes.

Automatización de procesos mediante CI/CD

- Implementé pipelines de Integración Continua y Despliegue Continuo en GitHub Actions, reduciendo el tiempo de entrega de modelos.
- Automatizé etapas clave como pruebas unitarias, empaquetado, versionado, despliegue y documentación automática.
- Gestioné artefactos y trazabilidad usando GitHub Packages y registro de contenedores.

Incorporación de seguridad desde etapas tempranas (Shift Left Security)

- Integré herramientas como SonarQube, Fortify y Orca Security para análisis estático, dinámico y escaneo de contenedores.
- Configuré políticas de seguridad automatizadas para evitar vulnerabilidades y malas prácticas de configuración.
- Implementé controles de cumplimiento basados en OWASP, NIST y buenas prácticas de DevSecOps.

Automatización de infraestructura

- Diseñé módulos de infraestructura como código (IaC) con Terraform desplegando con Terraform Cloud.
- Implementé aprovisionamiento automatizado de clusters de Databricks para ejecución de modelos y ADLS para almacenamiento seguro de datos.
- Integré HashiCorp Vault para la gestión centralizada de secretos, tokens y credenciales.

Trabajo colaborativo e impacto organizacional

- Coordiné mesas técnicas con los equipos de desarrollo, operaciones, arquitectura y seguridad.
- Elaboré documentación técnica y manuales operativos para transferir conocimiento al equipo.
- Lideré capacitaciones en prácticas DevSecOps, promoviendo la adopción y cultura de automatización segura.

CAPÍTULO III
ANÁLISIS DE CONTRIBUCIONES Y BENEFICIOS OBTENIDOS

Implementando el marco DevSecOps en proyectos de MLOps, apliqué mis capacidades técnicas en seguridad, infraestructura como código, automatización y arquitectura en la nube. Con herramientas como GitHub Actions, Vault, Artifactory, Fortify, Orca, Terraform y SonarQube, que me facilitaron la conversión de procesos manuales y lentos a flujos automatizados, seguros y auditable. Esto me posibilitó corroborar que la ingeniería no solamente consiste en tecnología, sino también en la concepción de soluciones sostenibles y replicables.

Asimismo, emplee habilidades de comunicación efectiva y liderazgo técnico. Además, promover la cultura de "seguridad desde el principio", guiar la adopción del modelo nuevo y aclarar ideas complejas a equipos de diferentes disciplinas. Esa interacción me permitió desarrollar la habilidad de administrar proyectos con una visión estratégica y no solo operativa.

3.1. ANÁLISIS DE SU CONTRIBUCIÓN EN TÉRMINOS DE LAS COMPETENCIAS Y HABILIDADES ADQUIRIDAS DURANTE SU FORMACIÓN PROFESIONAL. EXPLICAR SI SU CONTRIBUCIÓN REQUIRIÓ LA CONSULTA A OTRAS FUENTES DE INFORMACIÓN

a. Tecnología utilizada para el desarrollo de software e infraestructura
    i. Se aplicó arquitectura basada en MLOps y CI/CD.
    ii. Se integraron herramientas DevSecOps con plataformas de datos.
    iii. Se aplicó IaC con Terraform desplegando desde Terraform Cloud.
b. Seguridad de la información
    i. Se incorporaron controles de seguridad automatizados (vulnerabilidades).
    ii. Se implementó gestión centralizada de secretos (Hashicorp Vault).
    iii. Se aplicó Ethical Hacking end to end
c. Gestión y documentación
    i. Se elaboraron lineamientos y manuales operativos documentando los procesos operativos.

ii. Se definieron flujos de aprobaciones de ejecución dentro del pipeline, además para los esquemas de Pull Request (ciclo de versión de código fuente)
iii. Se documentó la arquitectura del proceso, flujo y tecnología empleada.

d. Comunicación y liderazgo técnico
    i. Se guiaron a equipos multidisciplinarios (datos, seguridad, operaciones, devsecops).
    ii. Se explicaron beneficios de CI/CD a usuarios no técnicos.

e. Análisis de problemas reales
    i. Se identificaron cuellos de botella de despliegue.
    ii. Se diferenciaron problemas técnicos versus problemas de proceso.
    iii. Se priorizaron soluciones que generen impacto positivo

3.2.EXPLICAR EL NIVEL DE BENEFICIO OBTENIDO POR EL CENTRO LABORAL DE SU CONTRIBUCIÓN A LA SOLUCIÓN DE LAS SITUACIONES PROBLEMÁTICAS

a. Reducción del tiempo de despliegue de Modelos
    i. De ciclos prolongado (meses), a despliegue en días de modelos de Machine Learning.
    ii. Garantizando la calidad, seguridad y estándar de cada desarrollo.

b. Mejor control de seguridad
    i. Centralización de secretos en proyectos críticos.
    ii. Código escaneado en fase desarrollo.
    iii. Alertas ante vulnerabilidades críticas/medias/bajas.

c. Mayor trazabilidad y cumplimiento
    i. Registro de usuarios de despliegue, versión y ambiente.
    ii. Sincronización con prácticas Zero Trust.

d. Estandarización de proyectos
    i. El framework queda como plantilla institucional.
    ii. Otros equipos pueden reutilizar los pipelines.
    iii. Se reduce la curva de aprendizaje.

e. Mejor percepción del área de TI
    i. Confianza de negocio en función de los beneficios otorgados
    ii. Se demuestra los beneficios y agilidad que tiene la nube.

Otra parte clave del proyecto fue la incorporación de herramientas que ayudaran a cuidar la seguridad de principio a fin. No se trató solo de agregar controles, sino de integrar plataformas que trabajaran junto con los desarrolladores en cada etapa. SonarQube, por ejemplo, se convirtió en un aliado diario: analizaba el código apenas se hacía un commit y mostraba de inmediato si había fallas o si se cumplían los estándares esperados. Esta retroalimentación rápida evitó muchos retrabajos y permitió mantener un nivel de calidad constante.

Por otro lado, Fortify se centró en revisar archivos específicos para detectar vulnerabilidades más profundas, sobre todo en los que estaban vinculados a MLOps y al uso de Python. Su integración fue clave para entender cómo se comportaban los modelos y qué tan seguros eran antes de darles uso real. Finalmente, Orca Security ayudó a revisar contenedores y recursos en la nube, anticipándose a posibles riesgos y frenando cualquier elemento que pudiera comprometer la seguridad antes de llegar a producción.

En conjunto, estas herramientas formaron una capa de protección que no solo respondió a incidentes, sino que permitió actuar antes de que ocurrieran. Con esto, fue posible construir un proceso ordenado, confiable y alineado con la idea de que la seguridad debe estar presente desde el diseño y no como un agregado final.

CAPÍTULO IV
CONCLUSIONES Y RECOMENDACIONES

El desarrollo de este proyecto permitió comprobar que aplicar un enfoque DevSecOps dentro de un entorno MLOps no es solo una buena práctica, sino prácticamente una necesidad cuando se quiere trabajar con modelos de machine learning de manera eficiente y confiable. La automatización, la seguridad integrada y los procesos estandarizados demostraron que es posible reducir tiempos y mejorar resultados sin sacrificar estabilidad ni protección.

4.1. CONCLUSIONES

Al adoptar DevSecOps, quedó claro que un enfoque preventivo de seguridad y una buena automatización pueden cambiar por completo la forma tradicional de construir tecnología. La integración de CI/CD, el uso de infraestructura como código y los despliegues más rápidos mostraron que es posible trabajar con fluidez y mantener un entorno controlado al mismo tiempo. Además, la aplicación de principios de Zero Trust permitió un mejor seguimiento de los modelos, garantizando que cada etapa cumpliera los requisitos de seguridad y calidad.

Este cambio también impulsó una forma de trabajo mucho más colaborativa entre los equipos y abrió la puerta a que otros proyectos puedan reutilizar plantillas o flujos ya probados. Gracias a eso, se comenzó a crear una cultura de mejora continua que elevó el nivel técnico del área de TI y dejó una buena impresión frente al negocio.

Como siguiente paso, se sugiere seguir ampliando este enfoque hacia más proyectos de análisis avanzado, reforzar el monitoreo continuo y reducir al mínimo las tareas manuales. También es clave seguir profundizando la adopción de Zero Trust para mantener ambientes productivos más resilientes y preparados para escalar.

4.2. RECOMENDACIONES

Para continuar con los resultados obtenidos, lo ideal es hacer que el modelo DevSecOps se extienda progresivamente a más áreas del desarrollo y análisis de datos. Así, el flujo de trabajo se vuelve más confiable, auditable y estable, incluso ante picos de demanda. Para lograrlo, es indispensable contar con un sistema de monitoreo que reúna métricas, logs y datos en tiempo real, lo que permitiría anticiparse a problemas y actuar de forma preventiva.

También es conveniente darle más peso a la seguridad mediante políticas basadas en Zero Trust y mecanismos automáticos que se ejecuten sin intervención manual. La autenticación fuerte, la renovación constante de credenciales y los controles de cumplimiento pueden elevar mucho el nivel de protección de cada capa del entorno tecnológico.

Otro punto relevante es fomentar el uso de infraestructura como código entre los equipos técnicos. Esta práctica reduce errores humanos, estandariza la creación de entornos y acelera los despliegues. A largo plazo, esto impacta directamente en la eficiencia operativa y en la capacidad para detectar anomalías en producción.

Por último, para manejar mejor los recursos en la nube, se recomienda incorporar principios de FinOps dentro del ciclo de vida MLOps. A medida que el uso computacional crece, es vital entender el gasto real, evaluar el retorno de inversión y aplicar estrategias que permitan equilibrar costos, rendimiento e innovación sin frenar el avance del proyecto.

REFERENCIAS BIBLIOGRÁFICAS

Databricks. (2024). MLOps and machine learning lifecycle management. https://docs.databricks.com

Datacenters.com. (2024). Kyndryl Peru Data Center – Overview técnico. Datacenters. https://www.datacenters.com/kyndryl-peru

GitHub. (2024). GitHub Actions Documentation https://docs.github.com/actions

Google Cloud. (2024). Continuous integration and continuous delivery for ML systems. https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-ml

HashiCorp. (2024). Vault documentation https://developer.hashicorp.com/vault

IBM. (2023). MLOps: Integrating ML models into production environments. IBM Cloud Blog https://www.ibm.com/cloud/blog/mlops

Kyndryl. (2025). Servicios de infraestructura crítica y modernización tecnológica. Kyndryl Perú. https://www.kyndryl.com/pe/es

Microsoft. (2024). Prácticas de MLOps para operaciones de aprendizaje automático. Microsoft Learn. https://learn.microsoft.com/azure/architecture/data-science-process/mlops

Microsoft News. (2024). BCP invierte más de S/ 2500 millones en modernización tecnológica junto a Kyndryl y Microsoft. https://news.microsoft.com/es-xl/bcp-invierte-mas-de-s-2500-millones-para-modernizar-su-infraestructura-tecnologica-de-la-mano-de-microsoft-y-kyndryl/

Red Hat. (2024). ¿Qué es DevSecOps y por qué es importante? Red Hat Blog. https://www.redhat.com/es/topics/devops/what-is-devsecops

Terraform by HashiCorp. (2024). Infrastructure as Code Documentation. https://developer.hashicorp.com/terraform

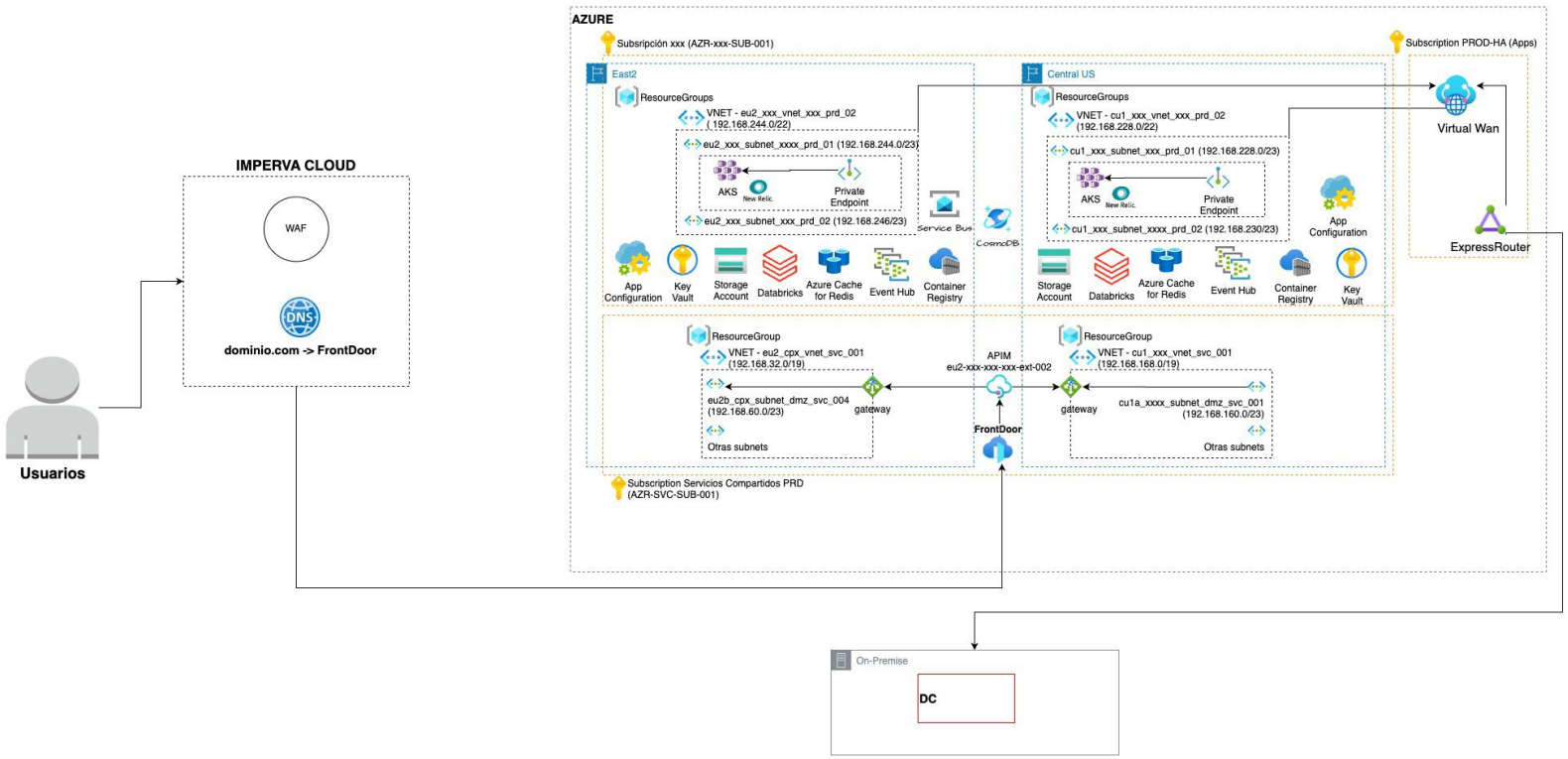
ANEXOS

ANEXO Nº 1: ARQUITECTURA BASE DE LA SOLUCIÓN CLOUD

La arquitectura que se implementó sigue un modelo multirregión, lo que significa que, si una zona sufre una caída o se presenta una falla inesperada, otra región puede asumir la carga casi de inmediato. Esta distribución no solo mejora la continuidad del servicio, sino que también aporta tranquilidad, porque el sistema está preparado para reaccionar ante imprevistos sin detener las operaciones.

Dentro del esquema se pueden identificar varios componentes clave. Uno de los más importantes es Azure Databricks, que se utilizó para desplegar los modelos del proyecto MLOps a través de pipelines de CI/CD. Gracias a esta integración, los modelos pudieron pasar desde el desarrollo hasta producción de forma automatizada y controlada.

También se destacan los servicios de almacenamiento como los Storage Accounts, que cumplieron un rol esencial. Allí se guardaron catálogos, librerías, archivos .jar, scripts .sh y otros recursos necesarios para ejecutar los init_scripts y para mantener los entornos configurados y listos para cada despliegue. En conjunto, esta arquitectura permitió trabajar de manera ordenada, resiliente y preparada para escalar cuando sea necesario.


AZURE
Subsripción xxx (AZR-xxx-SUB-001)
Subscription PROD-HA (Apps)
East2
ResourceGroups
VNET - eu2_xxx_vnet_xxx_prd_02 (192.168.244.0/22)
eu2_xxx_subnet_xxxx_prd_01 (192.168.244.0/23)
AKS
New Relic.
Private Endpoint
eu2_xxx_subnet_xxx_prd_02 (192.168.246/23)
Service Bus
CosmoDB
App Configuration
Key Vault
Storage Account
Databricks
Azure Cache for Redis
Event Hub
Container Registry
Central US
ResourceGroups
VNET - cu1_xxx_vnet_xxx_prd_02 (192.168.228.0/22)
cu1_xxx_subnet_xxx_prd_01 (192.168.228.0/23)
AKS
New Relic.
Private Endpoint
cu1_xxx_subnet_xxxx_prd_02 (192.168.230/23)
App Configuration
Storage Account
Databricks
Azure Cache for Redis
Event Hub
Container Registry
Key Vault
Virtual Wan
ExpressRouter
IMPERVA CLOUD
WAF
DNS
dominio.com -> FrontDoor
Usuarios
ResourceGroup
VNET - eu2_cpx_vnet_svc_001 (192.168.32.0/19)
eu2b_cpx_subnet_dmz_svc_004 (192.168.60.0/23)
gateway
Otras subnets
APIM
eu2-xxx-xxx-xxx-ext-002
FrontDoor
ResourceGroup
VNET - cu1_xxx_vnet_svc_001 (192.168.168.0/19)
gateway
cu1a_xxxx_subnet_dmz_svc_001 (192.168.160.0/23)
Otras subnets
Subscription Servicios Compartidos PRD (AZR-SVC-SUB-001)
On-Premise
DC

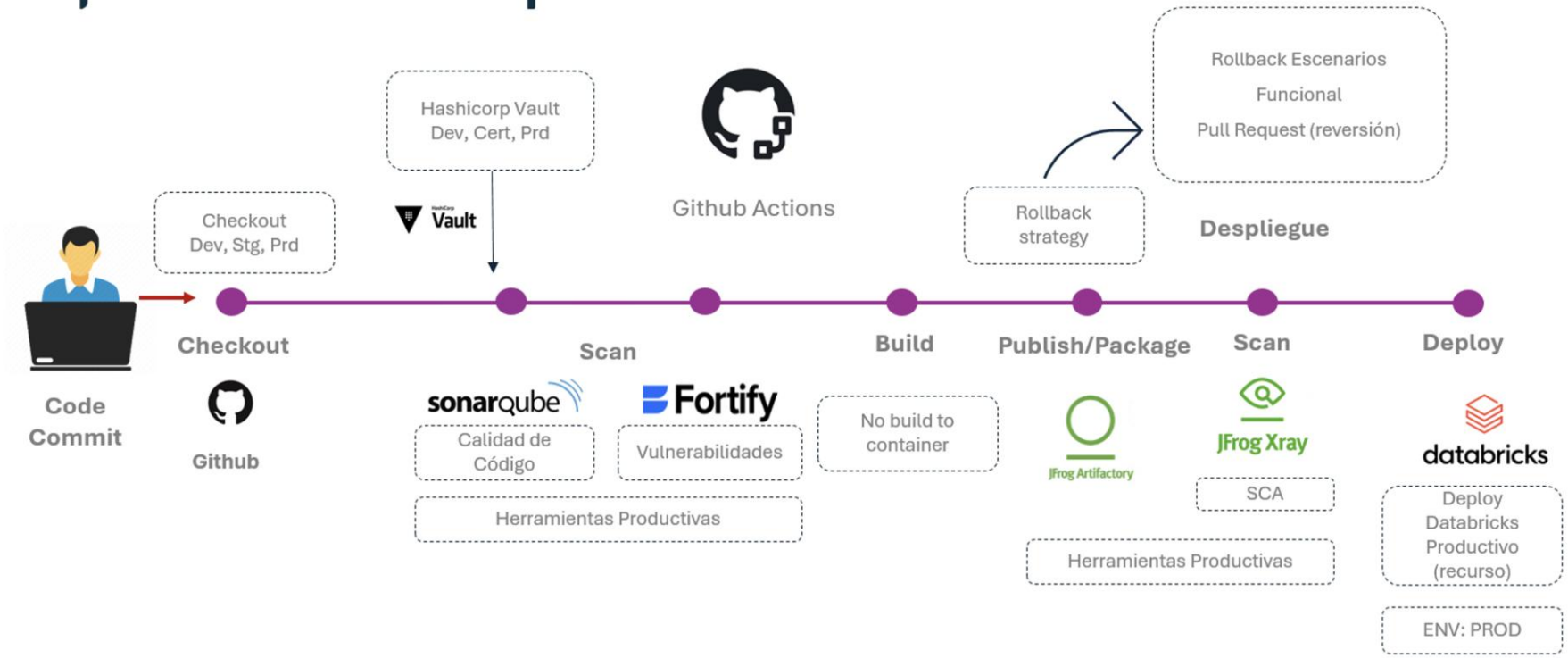
ANEXO Nº 2: PIPELINE DEVSECOPS IMPLEMENTADO

En este proyecto se aplicó un flujo DevSecOps que acompaña al desarrollo desde el primer paso: el commit del desarrollador. Desde ese momento, todo el ciclo de vida del software entra en acción de manera automatizada. Cada etapa tiene una herramienta específica que cumple su función y asegura que el código avance con seguridad y calidad.

Para proteger los secretos y credenciales se utilizó Hashicorp Vault, lo que evitó exponer información sensible en el código. Luego, con SonarQube, se revisó el código fuente buscando errores o malas prácticas antes de que generaran problemas mayores. Más adelante, Fortify se encargó de analizar vulnerabilidades más profundas, mientras que XRay revisó los paquetes y dependencias utilizadas en busca de riesgos ocultos. Finalmente, cuando todo estaba validado, el despliegue se realizaba en Azure Databricks, donde se ejecutaban los modelos y procesos del proyecto MLOps.

Así, la seguridad no fue una etapa adicional, sino parte del recorrido completo desde el momento en que se escribe el código hasta su despliegue final.

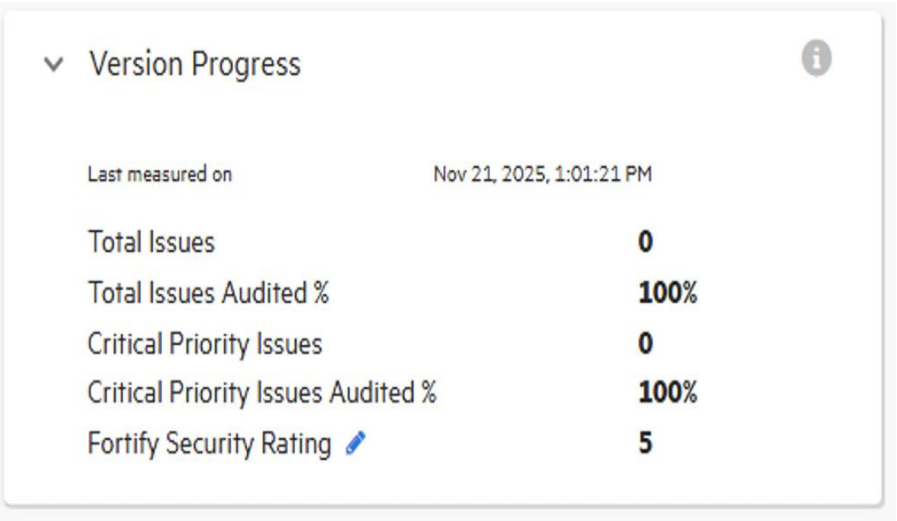
Flujo DevSecOps



ANEXO Nº 3: RESULTADOS DE ANÁLISIS DE VULNERABILIDADES

En la etapa de análisis de vulnerabilidades se utilizaron los reportes generados por Fortify, lo que permitió identificar riesgos presentes en el código antes de avanzar hacia el despliegue. Una vez que el escaneo terminó, los resultados se visualizaron directamente desde GitHub Actions, donde se puede ver el detalle de la ejecución y los hallazgos obtenidos. Además, en la otra vista se muestran los quality gates, que funcionan como un filtro previo y ayudan a validar que el código cumple con los estándares establecidos antes de continuar con el flujo de integración. Esta combinación de herramientas permitió revisar el código desde distintos ángulos y tomar decisiones con información más precisa.

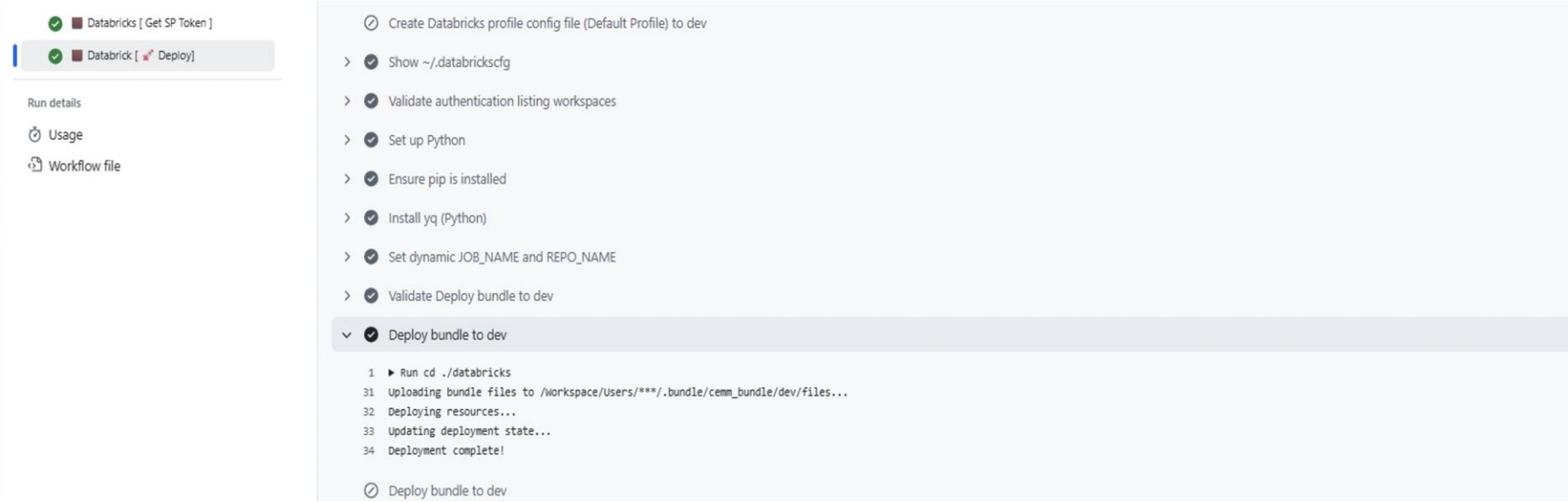
```
306  ▶ Run echo "Getting vulnerability counts for
359  Getting vulnerability counts for version ID:
360  No data
361
362  SSC aún procesando resultados, esperando 10s.
363  No data
364
365  SSC aún procesando resultados, esperando 10s.
366  No data
367
368  SSC aún procesando resultados, esperando 10s.
369  No data
370
371  SSC aún procesando resultados, esperando 10s.
372  No data
373
374  SSC aún procesando resultados, esperando 10s.
375  ▶ Run if [[ -z "0" ]]; then
428
429  # Análisis SAST Fortify para python
430  ### 🔴 Vulnerabilidades CRITICAL: 0
431  ### 🟠 Vulnerabilidades HIGH    : 0
432  ### 🟡 Vulnerabilidades MEDIUM  : 0
433  ### 🟢 Vulnerabilidades LOW     : 0
```



ANEXO Nº 4: EVIDENCIAS DE DESPLIEGUES AUTOMATIZADOS

Para este proyecto se configuró una ejecución completa de CI/CD mediante GitHub Actions, lo que permitió automatizar todo el proceso desde el desarrollo hasta el despliegue final. El flujo se inicia apenas el desarrollador realiza un commit en una rama del tipo feature/*, lo que activa automáticamente las diferentes etapas del pipeline. Gracias a esto, cualquier alerta o problema puede detectarse desde el comienzo del ciclo de desarrollo, evitando que los errores avancen a fases más críticas. Esta visibilidad temprana fue clave para mantener un proceso ordenado y con mayor control sobre la calidad del código.

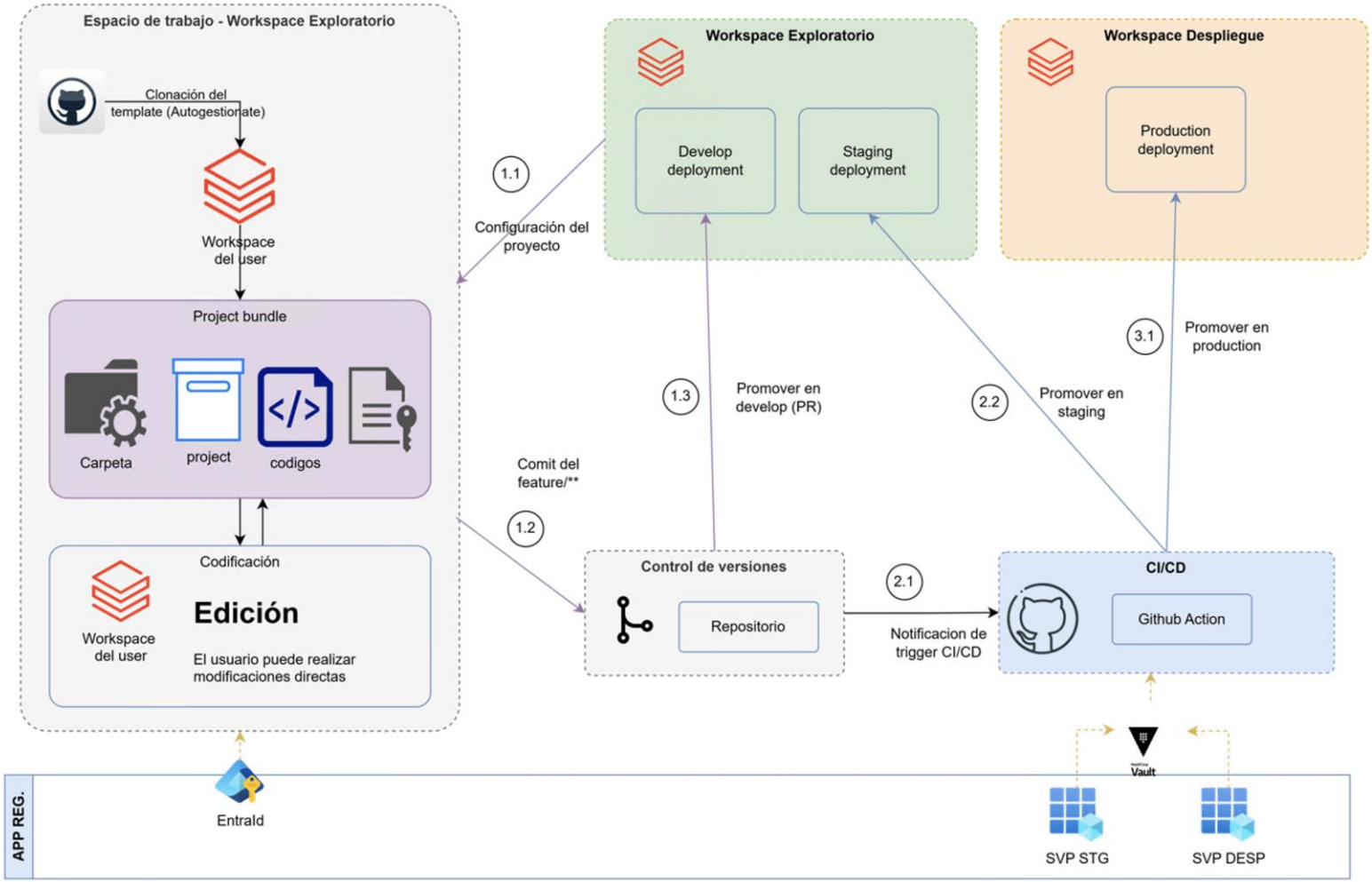
En esta etapa del pipeline se ejecutó la fase de despliegue continuo (CD), donde se puede ver claramente cómo el componente es enviado y publicado en Databricks. El resultado de la ejecución confirma que el bundle fue trasladado al entorno correspondiente y que el proceso finalizó de forma exitosa, indicando que la automatización está funcionando como se esperaba.



ANEXO Nº 5: FLUJO DE TRABAJO MEDIANTE ESQUEMA TRUNK BASE DEVELOPMENT

En este esquema se utilizó un flujo de desarrollo basado en el enfoque trunk-based, pensado específicamente para trabajar con modelos de manera ordenada y segura. La idea es sencilla: todo parte desde la rama develop, y a partir de ella se crean ramas del tipo feature/*, donde se realiza la codificación inicial. Una vez que cada cambio está listo, se abre un pull request para enviar esos cambios a las ramas que se utilizan para los despliegues.

Este proceso permite tener un control claro del código y facilita que los modelos puedan avanzar de forma gradual hacia los diferentes ambientes dentro de Databricks. Con este flujo se evita trabajar directamente sobre ramas sensibles, se promueve la revisión colaborativa y se asegura que cada versión pase por un proceso controlado antes de llegar a producción.


Espacio de trabajo - Workspace Exploratorio
Clonación del template (Autogestionate)
Workspace del user
Project bundle
Carpeta
project
codigos
Codificación
Workspace del user
Edición
El usuario puede realizar modificaciones directas
Entrald
APP REG.
1.1
Configuración del proyecto
Workspace Exploratorio
Develop deployment
Staging deployment
Workspace Despliegue
Production deployment
1.3
Promover en develop (PR)
2.2
Promover en staging
3.1
Promover en production
Comit del feature/**
1.2
Control de versiones
Repositorio
2.1
Notificacion de trigger CI/CD
CI/CD
Github Action
Vault
SVP STG
SVP DESP

ANEXO Nº 6: MODULO REUSABLE DE FORTIFY

Este módulo fue creado para conectarse con el servicio de Fortify y realizar el análisis de seguridad del código. Se le considera “reusable” porque no está pensado únicamente para los proyectos relacionados con MLOps, sino que también puede aplicarse en otros desarrollos estratégicos de la organización que necesiten este tipo de validaciones. De esta forma, se convierte en una pieza común que permite integrar Fortify en distintos repositorios sin tener que crear el mismo flujo desde cero cada vez.

```yaml
name: Fortify SCA SAST (Python)

on:
  push:
    branches: [ main ]
  pull_request:

jobs:
  fortify-sca-sast:
    # Recomendacion: tener un runner self-hosted con Fortify ya instalado
    runs-on: self-hosted

    env:
      FORTIFY_BUILD_ID: "python-mlops-app"
      FORTIFY_FPR_NAME: "fortify-python.fpr"

    steps:
      - name: Checkout repo
        uses: actions/checkout@v4

      - name: Set up Python
        uses: actions/setup-python@v5
        with:
          python-version: "3.11"    # Especifica versión

      # Instalar dependencias necesarias
      - name: Install dependencies
        run: |
          if [ -f requirements.txt ]; then
            pip install -r requirements.txt
          fi

      # (Opcional) Ejecutar tests antes del análisis
      - name: Run unit tests
        run: |
          if [ -d tests ]; then
            pytest || echo "Tests failed, but continuing for SAST..."
          fi

      # Limpia el build anterior de Fortify
      - name: Fortify - Clean previous build
        run: |
          sourceanalyzer -b "$FORTIFY_BUILD_ID" -clean || true

      # Análisis estático del código Python
      - name: Fortify - Translate Python source
        run: |
          # Ajusta la ruta del código si no es el root
          sourceanalyzer \
            -b "$FORTIFY_BUILD_ID" \
            -64 \
            -python "." \
            -verbose

      # Escaneo y generación de archivo .fpr
      - name: Fortify - Scan and create FPR
        run: |
          sourceanalyzer \
            -b "$FORTIFY_BUILD_ID" \
            -scan \
            -f "$FORTIFY_FPR_NAME" \
            -verbose

      # Publicar el .fpr como artefacto del pipeline
      - name: Upload Fortify FPR artifact
        uses: actions/upload-artifact@v4
        with:
          name: fortify-sca-report
          path: ${{ env.FORTIFY_FPR_NAME }}
```

ANEXO Nº 7: MODULO REUSABLE DE SONARQUBE

Este módulo fue diseñado para integrar SonarQube dentro del flujo de desarrollo y evaluar la calidad del código de forma automática. Se considera reutilizable porque no está limitado únicamente a los proyectos MLOps; puede implementarse también en otras iniciativas de la organización que necesiten validar su código mediante análisis estáticos. Gracias a este enfoque, distintos equipos pueden aprovechar la misma integración sin tener que construirla desde cero cada vez, lo que ahorra tiempo y estandariza el proceso de revisión.

```yaml
name: SonarQube SAST (Python)

on:
  push:
    branches: [ main ]
  pull_request:

jobs:
  sonarqube-scan:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout repository
        uses: actions/checkout@v4

      - name: Set up Python
        uses: actions/setup-python@v5
        with:
          python-version: "3.11"   # Ajusta la versión según tu proyecto

      - name: Install dependencies
        run: |
          if [ -f requirements.txt ]; then
            pip install -r requirements.txt
          fi

      - name: Run tests with coverage
        run: |
          if [ -d tests ]; then
            pip install pytest pytest-cov
            pytest --cov=. --cov-report=xml
          fi

      # Análisis con SonarQube Enterprise (self-managed)
      - name: SonarQube Scan
        uses: sonarsource/sonarqube-scan-action@v2
        env:
          SONAR_HOST_URL: ${{ secrets.SONAR_HOST_URL }}
          SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
        with:
          args: >
            -Dsonar.projectKey=python-mlops-devsecops
            -Dsonar.projectName=python-mlops-devsecops
            -Dsonar.sources=.
            -Dsonar.tests=tests
            -Dsonar.python.version=3.11
            -Dsonar.python.coverage.reportPaths=coverage.xml
            -Dsonar.exclusions=**/venv/**,**/tests/**

      # (Opcional) Bloquear si no pasa el Quality Gate
      - name: SonarQube Quality Gate
        uses: sonarsource/sonarqube-quality-gate-action@v1.1.0
        env:
          SONAR_HOST_URL: ${{ secrets.SONAR_HOST_URL }}
          SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
        with:
          scanMetadataReportFile: .scannerwork/report-task.txt
```

ANEXO Nº 8: MODULO REUSABLE DE DATABRICKS

Este módulo fue diseñado para ejecutar notebooks en Databricks como parte del pipeline de CI/CD. Se planteó como un componente reutilizable porque no está pensado únicamente para los flujos de MLOps, sino que puede integrarse también en otros proyectos estratégicos de la organización que necesiten automatizar despliegues o ejecuciones en Databricks. De esta manera, se evita reconstruir la misma lógica en cada repositorio y se logra un estándar que facilita la colaboración entre equipos y acelera los ciclos de desarrollo.

```yaml
name: Databricks CI - Run Notebook

on:
  push:
    branches: [ main ]
  pull_request:

jobs:
  databricks-notebook:
    runs-on: ubuntu-latest

    env:
      DATABRICKS_HOST: ${{ secrets.DATABRICKS_HOST }}
      DATABRICKS_TOKEN: ${{ secrets.DATABRICKS_TOKEN }}

    steps:
      - name: Checkout repo
        uses: actions/checkout@v4

      - name: Set up Python
        uses: actions/setup-python@v5
        with:
          python-version: "3.11"

      - name: Install dependencies
        run: |
          if [ -f requirements.txt ]; then
            pip install -r requirements.txt
          fi

      # (Opcional) tests locales antes de ir a Databricks
      - name: Run unit tests
        run: |
          if [ -d tests ]; then
            pytest
          fi

      # Ejecutar notebook en Databricks como un job one-shot
      - name: Run Databricks notebook
        uses: databricks/run-notebook@v0
        with:
          databricks-host: ${{ env.DATABRICKS_HOST }}
          databricks-token: ${{ env.DATABRICKS_TOKEN }}
          # ruta del notebook en el workspace
          notebook-path: "/Shared/mlops/notebooks/train_model"
          # o un cluster existente
          existing-cluster-id: ${{ vars.DATABRICKS_CLUSTER_ID }}
          # parámetros opcionales para el notebook
          notebook-params-json: '{"env": "dev", "run_id": "${{ github.run_id }}"}'
```

ANEXO Nº 9: ACTION DE CI/CD CONSUMIENDO TODOS LOS SERVICIOS MEDIANTE HASHICORP VAULT

Este módulo fue creado para conectarse con Hashicorp Vault y recuperar los secretos necesarios durante la ejecución del pipeline. Se diseñó con un enfoque reutilizable para que no solo sirva en proyectos de MLOps, sino también en otras iniciativas clave de la organización que requieran proteger credenciales o variables sensibles. Gracias a este esquema, los equipos pueden integrar Vault sin tener que configurar todo desde cero, manteniendo la seguridad centralizada y estandarizando la forma en que se consumen los secretos en diferentes repositorios.

```yaml
name: CI/CD Python
on:
  push:
    branches: [ main ]
  pull_request:

jobs:
  full-pipeline:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout repo
        uses: actions/checkout@v4

      # 1) Traer TODOS los secretos desde HashiCorp Vault
      - name: Load secrets from HashiCorp Vault
        id: vault
        uses: hashicorp/vault-action@v3
        with:
          url: https://vault.miempresa.com:8200
          token: ${{ secrets.VAULT_TOKEN }}
          caCertificate: ${{ secrets.VAULT_CA_CERT }}
          # path key | ENV_NAME ;
          secrets: |
            secret/data/ci/app        DB_PASSWORD    | DB_PASSWORD ;
            secret/data/ci/databricks HOST           | DATABRICKS_HOST ;
            secret/data/ci/databricks TOKEN          | DATABRICKS_TOKEN ;
            secret/data/ci/databricks CLUSTER_ID     | DATABRICKS_CLUSTER_ID ;
            secret/data/ci/sonar      HOST_URL       | SONAR_HOST_URL ;
            secret/data/ci/sonar      TOKEN          | SONAR_TOKEN ;
            secret/data/ci/fortify    TOKEN          | FORTIFY_TOKEN

      # 2) Setup de Python
      - name: Set up Python
        uses: actions/setup-python@v5
        with:
          python-version: "3.11"

      - name: Install dependencies
        run: |
          if [ -f requirements.txt ]; then
            pip install -r requirements.txt
          fi

      # 3) Tests usando secreto de DB desde Vault
      - name: Run unit tests (DB from Vault)
        env:
          DB_PASSWORD: ${{ env.DB_PASSWORD }}
        run: |
          echo "Conectando a la BD usando DB_PASSWORD desde Vault (no se imprime en logs)"
          # ejemplo:
          # pytest --db-password "${DB_PASSWORD}"

      # 4) Análisis con SonarQube usando secretos de Vault
      - name: SonarQube Scan
        if: env.SONAR_TOKEN != ''
        uses: sonarsource/sonarqube-scan-action@v2
        env:
          SONAR_HOST_URL: ${{ env.SONAR_HOST_URL }}
          SONAR_TOKEN: ${{ env.SONAR_TOKEN }}
        with:
          args: >
            -Dsonar.projectKey=python-mlops-devsecops
            -Dsonar.projectName=python-mlops-devsecops
            -Dsonar.sources=.
            -Dsonar.tests=tests
            -Dsonar.python.version=3.11
            -Dsonar.exclusions=**/venv/**,**/tests/**

      # 5) Análisis Fortify SCA usando FORTIFY_TOKEN de Vault (si aplica)
      - name: Fortify SCA Scan
        if: env.FORTIFY_TOKEN != ''
        env:
          FORTIFY_TOKEN: ${{ env.FORTIFY_TOKEN }}
        run: |
          echo "Ejecutando Fortify SCA usando credenciales desde Vault..."
          # aquí tus comandos reales de Fortify, por ejemplo:
          # sourceanalyzer -b python-mlops -clean
          # sourceanalyzer -b python-mlops -python "."
          # sourceanalyzer -b python-mlops -scan -f fortify-python.fpr

      # 6) Ejecutar notebook en Databricks usando HOST/TOKEN/CLUSTER_ID desde Vault
      - name: Run Databricks notebook (from Vault secrets)
        uses: databricks/run-notebook@v0
        env:
          DATABRICKS_HOST: ${{ env.DATABRICKS_HOST }}
          DATABRICKS_TOKEN: ${{ env.DATABRICKS_TOKEN }}
          DATABRICKS_CLUSTER_ID: ${{ env.DATABRICKS_CLUSTER_ID }}
        with:
          databricks-host: ${{ env.DATABRICKS_HOST }}
          databricks-token: ${{ env.DATABRICKS_TOKEN }}
          notebook-path: "/Shared/mlops/notebooks/train_model"
          existing-cluster-id: ${{ env.DATABRICKS_CLUSTER_ID }}
          notebook-params-json: '{"pipeline":"ci-cd","run_id":"${{ github.run_id }}"}'
```