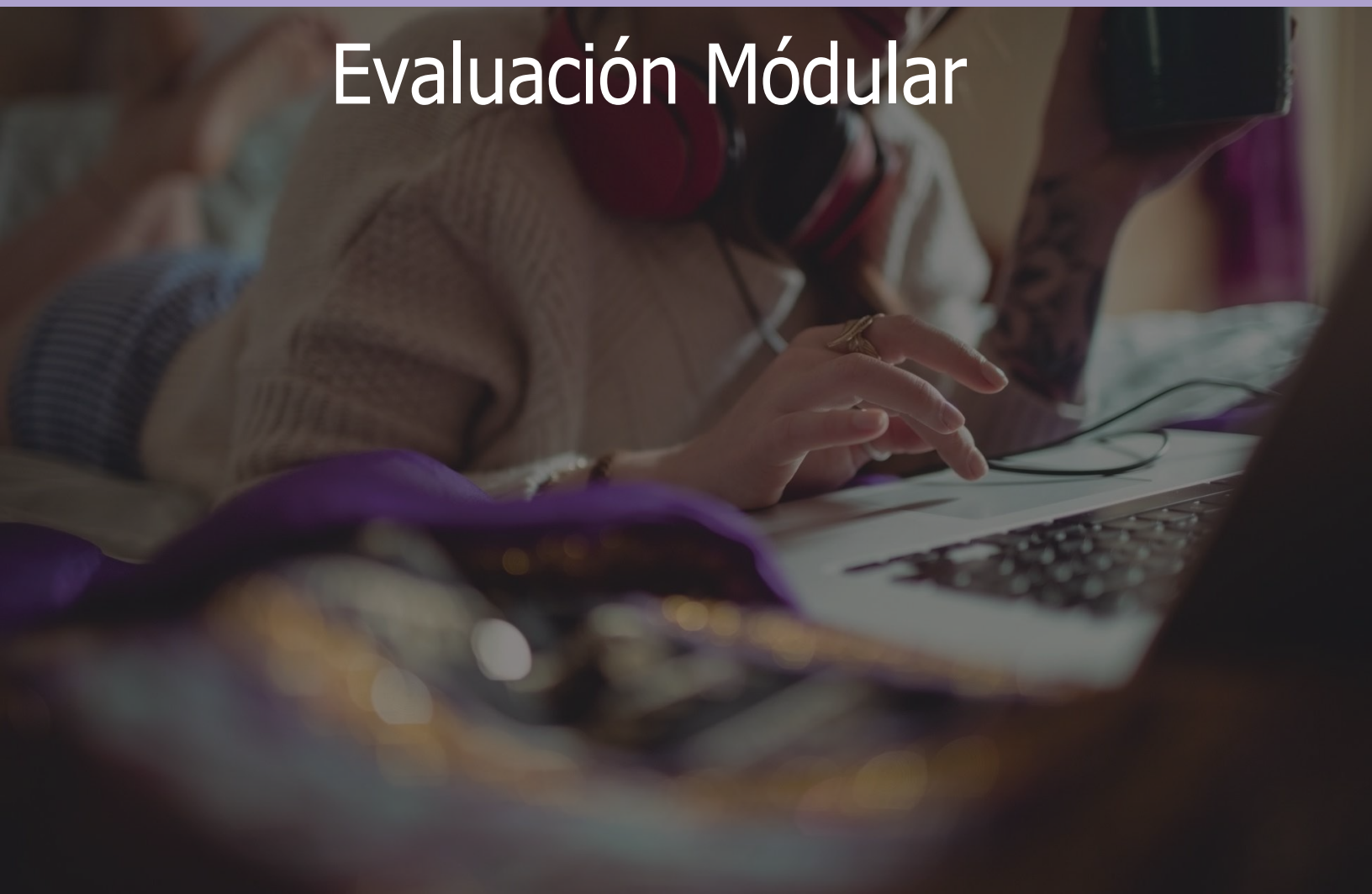


Módulo 10

Evaluación Modular



ACTIVIDAD:

MLOps en la Nube: Despliegue automatizado de un modelo predictivo real

Objetivo:

Desarrollar un sistema completo que integre un modelo de Machine Learning (clasificación o regresión), exponerlo como API REST mediante Flask, contenedorizado con Docker, y desplegado profesionalmente en una plataforma en la nube (AWS SageMaker, Google Vertex AI o Azure ML). Se espera automatizar al menos parte del flujo de trabajo usando buenas prácticas de CI/CD y demostrar la capacidad de monitoreo, versionado y actualización.

Contexto:

Has sido contratado por una fintech/startup de salud para integrar un modelo de predicción (puede ser de scoring de crédito o diagnóstico preventivo) dentro de su infraestructura cloud. Necesitan que este servicio sea:

- Escalable
- Fácil de actualizar
- Accesible vía REST
- Trazable y monitoreado

Deberás entregar un sistema funcional que esté desplegado y probado en un entorno en la nube.

Tiempo estimado de desarrollo: 120 minutos.

Formato de entrega: archivo comprimido .zip o .rar con código fuente y PDF.

Modalidad: grupal.



Requerimiento:

1. Modelo Predictivo

- Dataset a utilizar:
 - Salud: Breast Cancer (<https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data>)
- Entrenar y guardar el modelo usando joblib, pickle o .h5

2. API con Flask

- Crear una API con al menos dos rutas:
 - GET /: para probar el estado del servicio
 - POST /predict: para recibir un JSON y retornar predicción
- Validación de entradas, manejo de errores, logging

3. Dockerización

- Escribir un Dockerfile funcional que empaquete la API y modelo
- Incluir dependencias y entorno reproducible
- Probar localmente con docker run

4. CI/CD Automatizado (opcional, recomendado)

- Crear workflow en GitHub Actions o GitLab CI
- Automáticamente hacer:
 - Build del contenedor
 - Test básico de endpoints
 - Push a Docker Hub o Artifact Registry

5. Despliegue en la nube

- Elegir una plataforma:
 - AWS: SageMaker + S3
 - GCP: Vertex AI + Cloud Storage
 - Azure: Azure ML + Azure Container Instances
- Proceso completo: subida de modelo + deploy + test del endpoint
- Documentar con capturas o curl de prueba

6. Dashboard o monitoreo básico

- Mostrar cómo se monitorea el endpoint (CloudWatch, GCP Logs, Azure Insights)
- Bonus: establecer métricas (latencia, frecuencia de uso)





Rúbrica:

Indicador de logro / Criterio	Insuficiente (0%-20%)	Por lograrlo (21%-40%)	Medianamente logrado (41%-60%)	Logrado (61%-80%)	Sobresaliente (81%-100%)
Entrenamiento y serialización del modelo	No entrena modelo o no lo guarda.	Modelo entrenado, pero mal guardado o con errores.	Modelo entrenado y guardado correctamente, sin documentación clara.	Entrenamiento correcto, guardado con joblib/pickle/h5, y con prueba básica.	Entrenamiento optimizado, guardado y probado, con explicación clara del modelo y sus decisiones.
Desarrollo de API con Flask	No implementa API funcional.	API básica sin manejo de errores ni validación.	API funcional con rutas requeridas, validación limitada.	API funcional, bien estructurada, con validación, logging y manejo de errores.	API robusta, validada, bien documentada, con pruebas de endpoints (manuales o automatizadas).
Dockerización del sistema	No entrega Dockerfile o está incompleto.	Dockerfile presente, pero con errores o sin pruebas.	Dockerfile funcional, entorno reproducible, probado localmente.	Dockerfile correcto, imagen funcional, dependencias claras, probado con docker run.	Dockerfile optimizado, imagen documentada, uso eficiente de capas, pruebas locales exitosas.
Automatización CI/CD (opcional)	No implementa automatización.	Crea un workflow básico sin pruebas.	CI/CD con build funcional y push a registro.	CI/CD funcional con build, test de endpoints y push automatizado.	Flujo CI/CD completo y profesional con integración continua, pruebas automáticas y despliegue eficiente.
Despliegue en la nube	No realiza despliegue o lo hace fuera de la nube.	Despliegue incompleto o sin documentación.	Despliegue funcional, pero sin pruebas demostrables (capturas, curl).	Despliegue completo y probado en SageMaker, Vertex AI o Azure ML.	Despliegue profesional, documentado con pruebas efectivas, uso adecuado de servicios cloud y seguridad básica.
Monitoreo y dashboard	No presenta evidencia de monitoreo.	Muestra logs básicos sin análisis.	Integra monitoreo con herramienta cloud (CloudWatch, Logs).	Monitoreo funcional y documentado (errores, trazabilidad).	Monitoreo sólido con visualización de métricas (latencia, tráfico), capturas o dashboards.
Documentación y entrega	Entrega incompleta, desordenada o ilegible.	Código entregado sin explicación ni estructura.	Proyecto funcional, documentación básica (README, PDF).	Entrega ordenada con README claro, PDF explicativo y capturas.	Entrega profesional, documentación clara y completa, buenas prácticas y reflexiones finales.