LABORATORIO DE INTELIGENCIA ARTIFICIAL · IBERO

Módulo 3
Desarrollo, Despliegue y Automatización de Modelos IA

Plan de estudios detallado por tema, objetivo y práctica, con base en *Machine Learning Design Patterns* y *Kubeflow for Machine Learning*

40 h	5	124
DURACIÓN	TALLERES	TEMAS CON OBJETIVO

Saptiva AI

Para la Universidad Iberoamericana Ciudad de México

Versión 1.1
Noviembre 2026

Este documento detalla el Módulo 3 definido en el Project Charter de Ejecución del Laboratorio de IA. Las fechas por taller se publicarán al confirmarse el calendario del módulo.

RESUMEN DEL MÓDULO

Del notebook a producción, sobre las GPUs del laboratorio

El módulo capacita a académicos e investigadores en el ciclo completo de un modelo de aprendizaje profundo: entrenamiento, validación, registro, despliegue, monitoreo y automatización, usando la infraestructura de cómputo de la IBERO y el ecosistema Kubeflow. Cada tema tiene un objetivo verificable y se apoya en dos obras de referencia: los patrones de *Machine Learning Design Patterns* dan el «por qué» y *Kubeflow for Machine Learning* el «cómo» en la plataforma. Las prácticas usan datasets abiertos de Kaggle.

AUDIENCIA

5 administrativos + 20 académicos de áreas técnicas (Ingenierías, Ciencia de Datos)

PRERREQUISITOS

Python básico, uso de Jupyter y nociones de aprendizaje automático

ACCESO

Cuenta institucional (SSO) y VPN para sesiones virtuales

Cada bloque indica minutos por tema, objetivo de aprendizaje y referencia bibliográfica. Los bloques prácticos especifican dataset, componentes de Kubeflow y criterio de logro. Las sesiones se graban, editan y sistematizan conforme al Project Charter.

Distribución de horas



	TALLER	HORAS	TECNOLOGÍA PRINCIPAL
A	**Fundamentos de MLOps**	8 h	Kubeflow Pipelines v2, Kubeflow Notebooks, Kubeflow Hub (Model Registry)
B	**Entrenamiento en GPU**	12 h	Kubeflow Trainer v2 (TrainJob), Kubeflow SDK (TrainerClient), Katib
C	**Orquestación de Trabajos**	8 h	Kueue, Saptiva frIdA, Kubeflow Trainer v2
D	**Despliegue de Modelos**	6 h	KServe (InferenceService), Kubeflow Hub, Kubeflow Pipelines v2
E	**Optimización y Monitoreo**	6 h	Prometheus + Grafana, NVIDIA DCGM Exporter, Métricas de KServe

Hilo conductor. Un mismo modelo de clasificación de imágenes recorre los cinco talleres: se construye su pipeline (A), se entrena y optimiza en GPU (B), se ejecuta a través de la cola compartida (C), se publica como servicio (D) y se monitorea hasta disparar su reentrenamiento (E). Un dataset tabular complementa los patrones de representación de datos y drift.

FUNDAMENTOS

Bibliografía aplicada al módulo

Machine Learning Design Patterns (Lakshmanan, Robinson y Munn, 2020) cataloga 30 patrones; el módulo trabaja 21 de ellos. La notación **DPnn** en las tablas remite al número de patrón del libro.

PATRÓN	NOMBRE	FAMILIA	BLOQUES
DP1	Hashed Feature	Representación de datos	A4
DP3	Feature Cross	Representación de datos	A4
DP5	Reframing	Representación del problema	B6
DP10	Rebalancing	Representación del problema	B2
DP11	Useful Overfitting	Entrenamiento	B2, B3
DP12	Checkpoints	Entrenamiento	B2, B4, B6
DP13	Transfer Learning	Entrenamiento	A3, B2, B3
DP14	Distribution Strategy	Entrenamiento	B4
DP15	Hyperparameter Tuning	Entrenamiento	B5, B6
DP16	Stateless Serving Function	Servicio resiliente	D1, D3
DP17	Batch Serving	Servicio resiliente	D1, D3
DP18	Continued Model Evaluation	Servicio resiliente	A4, E1, E3
DP19	Two-Phase Predictions	Servicio resiliente	D2
DP20	Keyed Predictions	Servicio resiliente	D2, D3
DP21	Transform	Reproducibilidad	A2, A3, D1, E1
DP22	Repeatable Splitting	Reproducibilidad	A2, A3
DP25	Workflow Pipeline	Reproducibilidad	A2, A4, E4
DP27	Model Versioning	Reproducibilidad	A2, A3, B6, E4
DP28	Heuristic Benchmark	IA responsable	A3, B6
DP29	Explainable Predictions	IA responsable	E1
DP30	Fairness Lens	IA responsable	E1

FUNDAMENTOS

Kubeflow for Machine Learning y datasets

Kubeflow for Machine Learning (Grant, Karau, Lublinsky, Liu y Filonenko, 2020) aporta la arquitectura y los conceptos de cada componente.

CAPÍTULO	TÍTULO	BLOQUES
Cap. 1	*Kubeflow: What It Is and Who It Is For*	A1
Cap. 2	*Hello Kubeflow*	A1, A3
Cap. 3	*Kubeflow Design: Beyond the Basics*	A1, A2, C1
Cap. 4	*Kubeflow Pipelines*	A2, A3, A4
Cap. 6	*Artifact and Metadata Store*	A2, A4
Cap. 7	*Training a Machine Learning Model*	B4
Cap. 8	*Model Inference*	D1
Cap. 10	*Hyperparameter Tuning and Automated Machine Learning*	B5, C3

> **Nota de vigencia.** El libro documenta Kubeflow 1.x de 2020 (Pipelines v1, KFServing, TFJob/PyTorchJob y Katib v1alpha3). Sus conceptos siguen vigentes, pero el código del módulo usa las APIs actuales: Kubeflow Pipelines v2, KServe, Kubeflow Trainer v2, Kubeflow SDK y Kubeflow Hub. Del mismo modo, los ejemplos de *Machine Learning Design Patterns* están en Keras y SQL y se implementan aquí en PyTorch sobre Kubeflow.

Datasets de práctica

DATASET	DESCRIPCIÓN	USO
Intel Image Classification Kaggle	Imágenes de escenas en 6 clases (~25 k, 150×150 px), con particiones de entrenamiento, prueba y predicción sin etiquetas.	Hilo conductor: A3, B, C, D, E
Telco Customer Churn Kaggle	~7 k clientes con variables categóricas y numéricas; clase positiva minoritaria.	A4, C3–C4, E3
Catálogo curado para proyecto Kaggle	Playground Series (tabular) y datasets de imágenes de tamaño moderado seleccionados por el instructor; o dataset propio del participante.	B6

Los datasets se precargan en el volumen compartido del laboratorio, de modo que las prácticas no dependen de acceso a Internet desde el clúster. La licencia de cada dataset se verifica en Kaggle antes de publicar modelos o derivados.

TALLER A · 8 HORAS

A Fundamentos de MLOps

¿Cómo convierto un experimento en un proceso reproducible?

Las organizaciones más avanzadas tratan sus modelos de IA como software: con versiones, pruebas automáticas y despliegues controlados. En este taller el participante convierte un notebook en un pipeline reproducible y registra su primer modelo versionado, aplicando los patrones de reproducibilidad de Machine Learning Design Patterns.

Patrones MLDP: DP1 · DP3 · DP13 · DP21 · DP22 · DP25 · DP27 · DP28

OBJETIVOS DEL TALLER

- Describir el ciclo de vida de un modelo y sus retos recurrentes.
- Descomponer un flujo de ML en componentes con entradas y salidas tipadas.
- Aplicar los patrones Workflow Pipeline, Transform, Repeatable Splitting y Model Versioning.
- Registrar un modelo con linaje completo hacia el run, los datos y el código que lo produjeron.

TECNOLOGÍA

Kubeflow Pipelines v2 · Kubeflow Notebooks · Kubeflow Hub (Model Registry) · Volúmenes (PVC) · kfp-kubernetes · PyTorch / torchvision

BLOQUES

1 A1 · ¿Qué es MLOps y por qué importa?

TEORÍA BÁSICA · 2 H · VIRTUAL

Objetivo del bloque. Comprender por qué un modelo no es un sistema y ubicar cada componente del laboratorio en el ciclo de vida.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
20′	**Deuda técnica oculta en sistemas de ML**	Identificar los componentes ajenos al modelo (datos, configuración, servicio, monitoreo) que concentran el riesgo operativo.	MLDP cap. 1; Sculley et al. (2015)
20′	**Ciclo de vida del modelo**	Enumerar las etapas datos → entrenamiento → validación → registro → despliegue → monitoreo y el artefacto que produce cada una.	MLDP cap. 8 «ML Life Cycle»
20′	**Retos comunes: calidad de datos, reproducibilidad, drift y escala**	Asociar cada reto con la etapa del ciclo donde se origina.	MLDP cap. 1
15′	**Niveles de madurez MLOps (0, 1 y 2)**	Ubicar un proyecto propio en un nivel y nombrar el siguiente paso.	Google Cloud (MLOps)
25′	**Kubeflow: qué es y para quién**	Explicar el papel de Notebooks, Pipelines, Hub, Trainer, Katib, Kueue y KServe, y cómo se organizan en un espacio de trabajo.	KfML caps. 1 y 3
20′	**Demo: pipeline completo sobre imágenes**	Observar el recorrido de punta a punta que el grupo construirá en el módulo.	KfML cap. 2

2 A2 · Reproducibilidad y CI/CD para modelos

TEORÍA AVANZADA · 2 H · VIRTUAL

Objetivo del bloque. Conocer los patrones que hacen que un resultado se pueda repetir y su implementación en Kubeflow Pipelines v2.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
20′	**Contenedores e imágenes inmutables**	Justificar por qué el entorno de ejecución es parte del modelo.	KfML cap. 3
15′	**Kubernetes a nivel conceptual**	Leer un manifiesto e identificar pod, namespace, volumen y requests de GPU.	KfML cap. 3
25′	**Workflow Pipeline y anatomía de KFP v2**	Descomponer un flujo en componentes (`@dsl.component`) con artefactos tipados Dataset, Model y Metrics.	MLDP DP25; KfML cap. 4
15′	**Transform**	Separar entradas, features y transformaciones para que el preprocesamiento sea idéntico al entrenar y al servir.	MLDP DP21
15′	**Repeatable Splitting**	Construir particiones deterministas por hash de un identificador en lugar de muestreo aleatorio.	MLDP DP22
15′	**Artefactos, metadatos, linaje y caché**	Rastrear qué datos, código y parámetros produjeron un artefacto.	KfML cap. 6
15′	**Model Versioning y Kubeflow Hub**	Distinguir RegisteredModel, ModelVersion y artefacto, y cuándo crear una versión nueva.	MLDP DP27; docs Kubeflow Hub

3 A3 · Tu primer pipeline MLOps

PRÁCTICA GUIADA · 2 H · VIRTUAL

Objetivo del bloque. Construir y ejecutar un pipeline de cinco pasos que termina en un modelo registrado.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
15′	**Notebook del módulo**	Crear un notebook con la imagen del módulo, una partición MIG y el volumen de datasets montado.	KfML cap. 2; docs Notebooks
25′	**Componente de ingesta con split repetible**	Implementar la ingesta y una partición por hash del nombre de archivo.	MLDP DP22
30′	**Preprocesamiento y entrenamiento**	Persistir la transformación y entrenar ResNet-18 con transfer learning (backbone congelado).	MLDP DP21, DP13
15′	**Evaluación contra un benchmark heurístico**	Comparar el modelo contra la clase mayoritaria y reportar Metrics.	MLDP DP28
20′	**Compilar, enviar e inspeccionar**	Enviar el pipeline con `kfp.Client` y leer DAG, logs y artefactos en la UI.	KfML cap. 4
15′	**Registro en Kubeflow Hub**	Registrar la versión con métricas y enlace al run de origen.	MLDP DP27

DATASET
Kaggle · Intel Image Classification (6 clases, ~25 k imágenes), subconjunto de 3 k imágenes

KUBEFLOW
Notebooks · KFP v2 · `kfp.kubernetes.mount_pvc` · Hub

CRITERIO DE LOGRO
Run en verde, accuracy superior al benchmark heurístico y versión 1 registrada con linaje.

4 A4 · Experimenta con el pipeline

PRÁCTICA AUTÓNOMA · 2 H · VIRTUAL

Objetivo del bloque. Usar el pipeline como instrumento de experimentación y comprobar que es independiente del dataset.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
25′	**Parametrización y comparación de runs**	Lanzar tres runs en un Experiment variando learning rate, epochs y fracción de datos, y compararlos.	KfML cap. 4
20′	**Compuerta de calidad**	Registrar el modelo solo si supera el benchmark más un umbral (`dsl.If`).	MLDP DP25, DP27
15′	**Caché de pasos**	Re-ejecutar el pipeline y explicar qué pasos se reutilizan y por qué.	KfML cap. 6
15′	**Recurring runs**	Programar una ejecución periódica y discutir qué debería disparar un reentrenamiento.	MLDP DP18
30′	**Extensión tabular**	Adaptar el pipeline a datos tabulares aplicando Feature Cross o Hashed Feature.	MLDP DP3, DP1
15′	**Puesta en común**	Comparar resultados entre participantes y defender una decisión de producción.	—

DATASET
Kaggle · Intel Image Classification (6 clases, ~25 k imágenes); Kaggle · Telco Customer Churn (~7 k clientes, tabular desbalanceado)

KUBEFLOW
KFP v2 (Experiments, Recurring Runs, caché) · Hub

CRITERIO DE LOGRO
Tres runs comparados, compuerta de calidad funcionando y pipeline tabular ejecutado.

Evidencia del participante. Pipeline de cinco componentes ejecutado, versión registrada en Kubeflow Hub con linaje y variante tabular funcionando.

TALLER B · 12 HORAS

B Entrenamiento en GPU

¿Cómo uso las GPUs del laboratorio de forma eficiente y a escala?

Entrenar en una laptop puede tomar días; con las GPUs del laboratorio, minutos. El participante pasa de entrenar en un notebook a enviar TrainJobs distribuidos y a buscar hiperparámetros con Katib, aplicando los patrones de entrenamiento de Machine Learning Design Patterns.

Patrones MLDP: DP5 · DP10 · DP11 · DP12 · DP13 · DP14 · DP15 · DP28

OBJETIVOS DEL TALLER

- Explicar por qué las GPUs aceleran el aprendizaje profundo y qué implica una partición MIG.
- Aplicar Useful Overfitting, Checkpoints, Transfer Learning y Distribution Strategy.
- Llevar un entrenamiento de notebook a TrainJob con reanudación.
- Diseñar un experimento de Hyperparameter Tuning en Katib y elegir con criterio de costo.

TECNOLOGÍA

Kubeflow Trainer v2 (TrainJob)

Kubeflow SDK (TrainerClient) · Katib · PyTorch · DDP · TensorFlow / Keras · NVIDIA MIG · DCGM · TensorBoard

BLOQUES

1 B1 · ¿Por qué GPUs para IA?

TEORÍA BÁSICA · 2 H · VIRTUAL

Objetivo del bloque. Entender el hardware del laboratorio lo suficiente para pedir los recursos adecuados.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
25′	**CPU vs GPU: latencia vs throughput, núcleos CUDA y Tensor Cores**	Explicar por qué las multiplicaciones de matrices se benefician del paralelismo masivo.	Docs NVIDIA CUDA
20′	**Tensores y CUDA desde PyTorch**	Mover modelos y datos a GPU y medir tiempos correctamente (sincronización).	Docs PyTorch
15′	**Precisión numérica: FP32, TF32, BF16 y FP8**	Elegir una precisión según memoria disponible y estabilidad numérica.	Docs NVIDIA
20′	**GPUs del laboratorio y particiones MIG**	Interpretar qué partición tiene asignada su notebook y cuáles son sus límites.	Docs NVIDIA MIG
15′	**PyTorch y TensorFlow**	Comparar ambos ecosistemas y su soporte en el laboratorio.	—
10′	**El ciclo típico de entrenamiento**	Describir el ciclo con descenso de gradiente estocástico.	MLDP cap. 4
15′	**Demo: CPU vs GPU**	Observar la diferencia de tiempo en el mismo entrenamiento.	—

2 B2 · Entrenamiento eficiente

TEORÍA AVANZADA · 2 H · VIRTUAL

Objetivo del bloque. Dominar los patrones y ajustes que determinan si una GPU trabaja o espera.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
20′	**Batch size, learning rate y epochs**	Predecir el efecto de cambiar cada uno y usar schedules de learning rate.	MLDP cap. 4
15′	**Useful Overfitting**	Usar el sobreajuste de un batch como prueba de sanidad del código.	MLDP DP11
15′	**Checkpoints y early stopping**	Guardar estado completo y reanudar sin perder progreso.	MLDP DP12
20′	**Transfer Learning**	Decidir qué capas congelar y cuáles ajustar según el tamaño del dataset.	MLDP DP13
20′	**Precisión mixta y cuellos de botella de I/O**	Diagnosticar si la GPU espera datos (DataLoader, NVMe vs almacenamiento compartido, shards).	Docs PyTorch
15′	**Monitoreo de GPU y métricas de entrenamiento**	Leer utilización, memoria y potencia, y detectar sobreajuste en las curvas.	Docs NVIDIA DCGM
15′	**Rebalancing y métricas para clases desbalanceadas**	Elegir métricas (F1, AUC-PR) y técnicas de rebalanceo.	MLDP DP10

3 B3 · Tu primer entrenamiento en GPU

PRÁCTICA GUIADA · 2 H · VIRTUAL

Objetivo del bloque. Entrenar un clasificador de imágenes en GPU y medir el efecto de cada optimización.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
10′	**Catálogo de labs «GYM»**	Abrir el lab con GPU asignada y verificar el dispositivo.	—
15′	**Prueba de sanidad**	Sobreajustar un solo batch hasta pérdida cercana a cero.	MLDP DP11
35′	**Entrenamiento base con transfer learning**	Entrenar ResNet-18 y seguir las curvas en TensorBoard.	MLDP DP13
30′	**CPU vs GPU vs precisión mixta**	Medir imágenes por segundo y utilización de GPU en las tres configuraciones.	—
20′	**Análisis de errores**	Leer la matriz de confusión e identificar las clases que se confunden.	—
10′	**Cierre**	Registrar hallazgos en la bitácora del lab.	—

DATASET
Kaggle · Intel Image Classification (6 clases, ~25 k imágenes), completo

KUBEFLOW
Notebooks con GPU (MIG) · TensorBoard · métricas DCGM

CRITERIO DE LOGRO
Tabla comparativa de throughput y accuracy en tres configuraciones.

4 B4 · PyTorch: del notebook al TrainJob

PRÁCTICA GUIADA · 2 H · PRESENCIAL

Objetivo del bloque. Convertir el entrenamiento interactivo en un trabajo reproducible, reanudable y distribuible.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
20′	**Función de entrenamiento autocontenida**	Refactorizar el código a una función con parámetros explícitos.	—
20′	**Checkpoints en volumen persistente**	Reanudar un entrenamiento tras una interrupción simulada.	MLDP DP12
30′	**Enviar un TrainJob**	Lanzar el entrenamiento con `TrainerClient` y el runtime `torch-distributed`.	Docs Trainer v2; KfML cap. 7
15′	**Seguimiento del TrainJob**	Consultar estado, logs y progreso desde el SDK y la UI.	Docs Trainer v2
25′	**Distribution Strategy**	Escalar con DDP de 1 a 2 GPUs y calcular speedup y eficiencia.	MLDP DP14; KfML cap. 7
10′	**Cierre**	Comparar costo y beneficio de distribuir.	—

DATASET
Kaggle · Intel Image Classification (6 clases, ~25 k imágenes)

KUBEFLOW
Trainer v2 (TrainJob, ClusterTrainingRuntime) · Kubeflow SDK · PVC

CRITERIO DE LOGRO
TrainJob completado, reanudado desde checkpoint y medición de speedup con 2 GPUs.

5 B5 · TensorFlow y optimización de hiperparámetros con Katib

PRÁCTICA AUTÓNOMA · 2 H · PRESENCIAL

Objetivo del bloque. Comparar frameworks y automatizar la búsqueda de hiperparámetros sin desperdiciar GPU.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
30′	**El mismo modelo en TensorFlow/Keras**	Reproducir el entrenamiento en Keras dentro del notebook y comparar API y rendimiento.	MLDP cap. 4 (ejemplos Keras)
15′	**Conceptos de Katib**	Distinguir Experiment, Suggestion, Trial, objetivo y métricas.	KfML cap. 10; MLDP DP15
35′	**Lanzar un Experiment**	Buscar learning rate, weight decay y capas congeladas con TPE, 12 trials y paralelismo 3.	KfML cap. 10
25′	**Early stopping y análisis de trials**	Aplicar median stopping e interpretar la tabla y las gráficas de trials.	Docs Katib
15′	**Elegir con criterio**	Seleccionar la configuración balanceando métrica y horas de GPU consumidas.	MLDP DP15

DATASET
Kaggle · Intel Image Classification (6 clases, ~25 k imágenes)

KUBEFLOW
Katib (Experiment, TPE, early stopping) · Kubeflow SDK

CRITERIO DE LOGRO
Experiment terminado y configuración elegida con justificación de costo.

6 B6 · Proyecto de entrenamiento con Kaggle

PRÁCTICA AUTÓNOMA · 2 H · PRESENCIAL

Objetivo del bloque. Aplicar el flujo completo de entrenamiento a un problema elegido por el participante.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
20′	**Planteamiento del problema**	Elegir dataset, métrica y benchmark heurístico; evaluar si conviene reformular el problema.	MLDP DP5, DP28
50′	**Entrenamiento**	Entrenar con TrainJob y checkpoints y, opcionalmente, una búsqueda en Katib.	MLDP DP12, DP15
10′	**Registro**	Registrar el mejor modelo en Kubeflow Hub (se usa en el Taller D).	MLDP DP27
30′	**Presentaciones**	Exponer en 5 minutos problema, métrica, resultado y costo de GPU.	—
10′	**Retrospectiva**	Identificar qué funcionó y qué no.	—

DATASET
Kaggle · catálogo curado (Playground Series y datasets de imágenes) o dataset propio

KUBEFLOW
Trainer v2 · Katib (opcional) · Hub

CRITERIO DE LOGRO
Modelo superior al benchmark heurístico, registrado y presentado.

Evidencia del participante. TrainJob con checkpoints, Experiment de Katib y modelo del proyecto registrado en Kubeflow Hub.

TALLER C · 8 HORAS

Orquestación de Trabajos

¿Quién usa la GPU primero y cómo se reparte de forma justa?

Cuando 20 investigadores quieren usar pocas GPUs a la vez, alguien tiene que decidir quién va primero. El laboratorio usa Kueue para admitir trabajos según cuotas, prioridades y fair-share, bajo las políticas de segregación de Saptiva frIdA que protegen el servicio de inferencia institucional.

Patrones MLDP: Plataforma: Kueue, frIdA · KfML cap. 3

OBJETIVOS DEL TALLER

- Distinguir admisión, scheduling y ciclo de vida de un trabajo.
- Explicar la jerarquía de colas, cuotas y prioridades del laboratorio.
- Enviar, monitorear, cancelar y re-encolar trabajos en su LocalQueue.
- Proponer políticas de uso justo con datos de tiempos de espera.

TECNOLOGÍA

Kueue · Saptiva frIdA · Kubeflow Trainer v2 · Katib · WorkloadPriorityClass · Particiones MIG

BLOQUES

1 C1 · El problema de los recursos compartidos

TEORÍA BÁSICA · 2 H · VIRTUAL

Objetivo del bloque. Entender por qué compartir GPUs requiere reglas explícitas y quién las aplica.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
20′	**Escasez de GPU**	Reconocer starvation, acaparamiento y GPUs ociosas en un laboratorio compartido.	—
25′	**Tres capas: admisión, scheduling y ciclo de vida**	Asignar cada responsabilidad a Kueue, al scheduler de Kubernetes o al controlador del trabajo.	Docs Kueue
25′	**FIFO, prioridades y fair-share**	Comparar políticas de asignación y sus efectos en la espera.	Docs Kueue
20′	**Segregación inferencia vs investigación**	Explicar cómo frIdA protege el chat institucional del entrenamiento pesado.	Charter §2.1
15′	**Trabajos interactivos vs batch**	Distinguir qué se encola (TrainJobs, trials) y qué no (notebooks).	KfML cap. 3
15′	**Demo: la cola en tiempo real**	Observar admisiones y esperas en vivo.	—

2 C2 · Colas, cuotas y prioridades en el laboratorio

TEORÍA AVANZADA · 2 H · VIRTUAL

Objetivo del bloque. Leer la configuración real de colas del laboratorio y predecir cuándo correrá un trabajo.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
20′	**ResourceFlavor y perfiles MIG**	Relacionar cada flavor con un tipo de partición o GPU completa.	Docs Kueue
20′	**ClusterQueue, LocalQueue y cuota nominal**	Leer la cuota de su espacio de trabajo y su pool.	Docs Kueue
20′	**Cohorts: préstamo de cuota**	Explicar cuándo un pool puede usar cuota ociosa de otro.	Docs Kueue
20′	**WorkloadPriorityClass y preemption**	Predecir qué trabajo se interrumpe cuando llega uno de mayor prioridad.	Docs Kueue
15′	**Gang scheduling y topología**	Explicar por qué un trabajo distribuido necesita todas sus GPUs a la vez.	Docs Kueue; Docs Trainer v2
10′	**Katib dentro de la cola**	Acotar el paralelismo de un Experiment mediante su LocalQueue.	Docs Katib
15′	**Buenas prácticas**	Dimensionar requests y liberar recursos al terminar.	—

3 C3 · Envía trabajos a la cola

PRÁCTICA GUIADA · 2 H · VIRTUAL

Objetivo del bloque. Operar la cola del laboratorio de punta a punta.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
15′	**Explorar la LocalQueue**	Consultar la cola de su espacio y la cuota disponible.	Docs Kueue
20′	**TrainJob encolado**	Enviar un TrainJob con la etiqueta `kueue.x-k8s.io/queue-name`.	Docs Kueue
20′	**Estados del Workload**	Seguir pendiente, cuota reservada, admitido y terminado, y leer los eventos.	Docs Kueue
30′	**Lote de trabajos tabulares**	Enviar diez trabajos de una búsqueda en malla y observar el orden de admisión.	—
25′	**Katib en la cola**	Apuntar los trials de un Experiment a la cola y observar el paralelismo efectivo.	KfML cap. 10
10′	**Cierre**	Registrar tiempos de espera en la bitácora.	—

DATASET
Kaggle · Telco Customer Churn (~7 k clientes, tabular desbalanceado); Kaggle · Intel Image Classification (6 clases, ~25 k imágenes)

KUBEFLOW
Kueue (LocalQueue, Workload) · Trainer v2 · Katib

CRITERIO DE LOGRO
Once o más trabajos admitidos y terminados vía cola, con tiempos registrados.

4 C4 · Gestiona tus propios trabajos

PRÁCTICA AUTÓNOMA · 2 H · VIRTUAL

Objetivo del bloque. Experimentar la contención real y convertirla en una propuesta de política.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
30′	**Saturación controlada**	Enviar trabajos en grupo y observar fair-share entre espacios.	—
20′	**Preemption por prioridad**	Observar cómo un trabajo de prioridad alta desplaza uno de taller.	Docs Kueue
30′	**Cancelar, re-encolar y redimensionar**	Medir el tiempo a admisión pidiendo particiones de distinto tamaño.	—
20′	**Análisis de la bitácora**	Calcular espera media y throughput del grupo.	—
20′	**Discusión de políticas**	Proponer reglas de uso justo para el laboratorio.	—

DATASET
Kaggle · Telco Customer Churn (~7 k clientes, tabular desbalanceado); Kaggle · Intel Image Classification (6 clases, ~25 k imágenes)

KUBEFLOW
Kueue · WorkloadPriorityClass · Trainer v2

CRITERIO DE LOGRO
Bitácora con tiempos de espera y propuesta de política argumentada.

Evidencia del participante. Bitácora de trabajos gestionados en la cola con análisis de espera y propuesta de política de uso justo.

TALLER D · 6 HORAS

D Despliegue de Modelos

¿Cómo convierto un modelo entrenado en un servicio que otros puedan usar?

Un modelo entrenado no sirve si nadie puede usarlo. El participante despliega el modelo del Taller B en línea y por lotes con KServe, aplicando los patrones de servicio resiliente de Machine Learning Design Patterns. Los LLMs institucionales se sirven en el Motor de Inferencias de Saptiva (vLLM) y se usan como referencia.

Patrones MLDP: DP16 · DP17 · DP19 · DP20 · DP21

OBJETIVOS DEL TALLER

- Distinguir servicio en línea y por lotes y elegir según el caso.
- Aplicar Stateless Serving Function, Batch Serving y Keyed Predictions.
- Desplegar un modelo registrado y consumirlo por API.
- Medir latencia y throughput y ejecutar un despliegue canary.

TECNOLOGÍA

KServe (InferenceService) · Kubeflow Hub · Kubeflow Pipelines v2 · Motor de Inferencias Saptiva (vLLM) · Open Inference Protocol v2 · Streamlit · Locust

BLOQUES

1 D1 · De modelo a servicio

TEORÍA BÁSICA · 2 H · VIRTUAL

Objetivo del bloque. Entender qué significa desplegar y cómo lo resuelve KServe.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
20′	**Inferencia en línea vs por lotes**	Elegir el modo de servicio según latencia requerida y volumen.	MLDP DP17
25′	**Stateless Serving Function**	Exportar el modelo con firma explícita y el preprocesamiento incluido.	MLDP DP16, DP21
15′	**Protocolos de inferencia**	Construir peticiones REST/gRPC con Open Inference Protocol v2.	Docs KServe
25′	**Arquitectura de KServe**	Describir InferenceService, predictor, transformer, ServingRuntime y storageUri.	KfML cap. 8; Docs KServe
15′	**Serverless vs despliegue estándar**	Decidir cuándo conviene escalar a cero.	Docs KServe
20′	**Servicio de LLMs**	Explicar por qué un LLM necesita un motor especializado como vLLM.	—

2 D2 · Arquitectura de inferencia

TEORÍA AVANZADA · 1 H · VIRTUAL

Objetivo del bloque. Razonar sobre rendimiento y resiliencia de un servicio de inferencia.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
15′	**Latencia p50/p95, throughput y SLOs**	Definir un SLO de latencia para un caso concreto.	—
10′	**Batching y concurrencia**	Predecir el efecto del batching en latencia y throughput.	Docs KServe
15′	**Autoescalado, canary y rollback**	Diseñar una liberación gradual con reversa.	Docs KServe
20′	**Two-Phase y Keyed Predictions**	Reconocer cuándo dividir la predicción en dos etapas y cuándo usar llaves.	MLDP DP19, DP20

3 D3 · Despliega tu primer modelo

PRÁCTICA GUIADA · 2 H · VIRTUAL

Objetivo del bloque. Publicar el modelo del Taller B en línea y por lotes.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
20′	**Exportar el modelo**	Exportar a TorchScript u ONNX con el preprocesamiento incluido.	MLDP DP16, DP21
25′	**InferenceService desde Kubeflow Hub**	Crear el servicio a partir de la versión registrada y verificar su estado.	Docs KServe, Hub
15′	**Probar el endpoint**	Enviar peticiones de prueba con Open Inference Protocol v2.	Docs KServe
25′	**Batch serving con llaves**	Clasificar las ~7 k imágenes sin etiqueta en un componente de pipeline, conservando el identificador.	MLDP DP17, DP20
25′	**Interfaz Streamlit**	Construir una interfaz mínima que consuma el endpoint.	—
10′	**Comparación con LLM institucional**	Consumir un modelo del Motor de Inferencias por API.	—

DATASET
Kaggle · Intel Image Classification (6 clases, ~25 k imágenes) (carpeta de predicción sin etiquetas)

KUBEFLOW
KServe · Hub · KFP v2 · Motor de Inferencias Saptiva

CRITERIO DE LOGRO
Endpoint respondiendo, archivo de predicciones por lotes con llaves e interfaz funcionando.

4 D4 · Prueba de carga y canary

PRÁCTICA AUTÓNOMA · 1 H · PRESENCIAL

Objetivo del bloque. Medir el servicio bajo carga y liberar una versión nueva de forma segura.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
25′	**Prueba de carga**	Generar carga con Locust y medir p50, p95 y peticiones por segundo.	—
10′	**Autoescalado**	Observar el número de réplicas frente a la carga.	Docs KServe
15′	**Canary y rollback**	Enviar 20 % del tráfico a la versión 2 y revertir.	Docs KServe
10′	**Discusión**	Proyectar cómo escalaría a producción real.	—

DATASET
Kaggle · Intel Image Classification (6 clases, ~25 k imágenes)

KUBEFLOW
KServe (canaryTrafficPercent) · Locust

CRITERIO DE LOGRO
Reporte de latencia bajo carga y canary ejecutado con rollback.

Evidencia del participante. Endpoint en línea, predicciones por lotes con llaves, interfaz Streamlit y reporte de carga.

TALLER E · 6 HORAS

E Optimización y Monitoreo

¿Cómo sé que mi modelo sigue funcionando y cómo lo hago más eficiente?

Un modelo en producción puede degradarse o consumir más recursos de los necesarios. El participante monitorea su servicio, detecta drift, compara un modelo optimizado y cierra el ciclo regresando al pipeline del Taller A, aplicando los patrones de evaluación continua e IA responsable.

Patrones MLDP: DP18 · DP21 · DP29 · DP30

OBJETIVOS DEL TALLER

- Distinguir métricas operativas de métricas de calidad del modelo.
- Aplicar Continued Model Evaluation y detectar drift con pruebas estadísticas.
- Diagnosticar cuellos de botella con dashboards y alertas.
- Decidir cuándo cuantizar, podar o reentrenar.

TECNOLOGÍA

Prometheus + Grafana · NVIDIA DCGM Exporter · Métricas de KServe · Evidently · PyTorch / ONNX Runtime · Kubeflow Pipelines v2

BLOQUES

1 E1 · ¿Por qué monitorear?

TEORÍA BÁSICA · 2 H · PRESENCIAL

Objetivo del bloque. Saber qué vigilar en un modelo desplegado y por qué se degrada.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
25′	**Data drift, concept drift y training-serving skew**	Distinguir los tres fenómenos con ejemplos.	MLDP cap. 1, DP21
20′	**Continued Model Evaluation**	Diseñar cómo obtener etiquetas tardías y evaluar en producción.	MLDP DP18
20′	**Métricas operativas vs de calidad**	Separar latencia, errores y saturación de precisión y calibración.	—
20′	**Pruebas estadísticas de drift**	Interpretar PSI, Kolmogorov–Smirnov y chi-cuadrada.	Docs Evidently
20′	**Explainable Predictions y Fairness Lens**	Incorporar explicabilidad y equidad al monitoreo.	MLDP DP29, DP30
15′	**Demo: dashboard del laboratorio**	Recorrer los paneles de GPU e inferencia.	—

2 E2 · Técnicas de optimización

TEORÍA AVANZADA · 1 H · PRESENCIAL

Objetivo del bloque. Conocer las palancas para servir más rápido y con menos recursos.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
20′	**Cuantización**	Comparar cuantización post-entrenamiento (INT8/FP8) y entrenamiento consciente de cuantización.	Docs PyTorch
15′	**Pruning y destilación**	Explicar qué se elimina o transfiere y su costo en precisión.	—
15′	**Batching y runtimes optimizados**	Ubicar ONNX Runtime y TensorRT en la cadena de servicio.	Docs ONNX Runtime
10′	**Matriz de decisión**	Elegir la técnica según latencia, costo y precisión.	—

3 E3 · Monitorea tu modelo desplegado

PRÁCTICA GUIADA · 2 H · PRESENCIAL

Objetivo del bloque. Instrumentar el servicio del Taller D y detectar drift en datos reales.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
25′	**Dashboards de Grafana**	Leer paneles de GPU (DCGM) y de KServe para su servicio.	—
25′	**Cuello de botella**	Identificar el recurso limitante del endpoint bajo carga.	—
30′	**Drift tabular**	Generar un reporte de Evidently entre datos de referencia y un lote con distribución desplazada.	MLDP DP18
20′	**Drift en imágenes**	Detectar el cambio de brillo y contraste mediante estadísticos y embeddings.	—
20′	**Alertas**	Configurar alertas de latencia p95 y de uso de GPU.	—

DATASET

Kaggle · Telco Customer Churn (~7 k clientes, tabular desbalanceado); Kaggle · Intel Image Classification (6 clases, ~25 k imágenes) con perturbaciones

KUBEFLOW

Grafana · DCGM · métricas KServe · Evidently

CRITERIO DE LOGRO

Dos reportes de drift y una alerta disparada en una simulación.

4 E4 · Diagnóstico, mejora y cierre del ciclo

PRÁCTICA AUTÓNOMA · 1 H · PRESENCIAL

Objetivo del bloque. Resolver un problema simulado y cerrar el ciclo MLOps.

MIN	TEMA	OBJETIVO: AL TERMINAR, EL PARTICIPANTE PODRÁ…	REFERENCIA
20′	**Diagnóstico**	Encontrar la causa de un problema de rendimiento o drift simulado.	—
20′	**Modelo original vs cuantizado**	Comparar latencia, tamaño y precisión.	—
15′	**Cierre del ciclo**	Relanzar el pipeline del Taller A y registrar una versión nueva.	MLDP DP25, DP27
5′	**Reflexión**	Resumir buenas prácticas de operación.	—

DATASET
Kaggle · Intel Image Classification (6 clases, ~25 k imágenes)

KUBEFLOW
KServe · KFP v2 · Hub

CRITERIO DE LOGRO
Nueva versión registrada como respuesta a una señal de monitoreo.

Evidencia del participante. Dashboard con alerta, reportes de drift, comparación de modelo optimizado y versión reentrenada.

REFERENCIAS

Bibliografía

Lakshmanan, V., Robinson, S. y Munn, M. (2020). *Machine Learning Design Patterns: Solutions to Common Challenges in Data Preparation, Model Building, and MLOps.* O'Reilly Media. ISBN 978-1-098-11578-4. BÁSICA

Grant, T., Karau, H., Lublinsky, B., Liu, R. y Filonenko, I. (2020). *Kubeflow for Machine Learning: From Lab to Production.* O'Reilly Media. ISBN 978-1-492-05012-4. BÁSICA

Sculley, D. et al. (2015). Hidden Technical Debt in Machine Learning Systems. *Advances in Neural Information Processing Systems 28*. COMPLEMENTARIA

Google Cloud. MLOps: Continuous delivery and automation pipelines in machine learning. Centro de arquitectura de Google Cloud. COMPLEMENTARIA

The Kubeflow Authors. Documentación oficial de Kubeflow Pipelines v2, Trainer v2, Katib, KServe, Notebooks y Kubeflow Hub. kubeflow.org/docs. TÉCNICA

Kubernetes SIGs. Documentación de Kueue. kueue.sigs.k8s.io. TÉCNICA

NVIDIA. Documentación de CUDA, Multi-Instance GPU (MIG) y DCGM. docs.nvidia.com. TÉCNICA