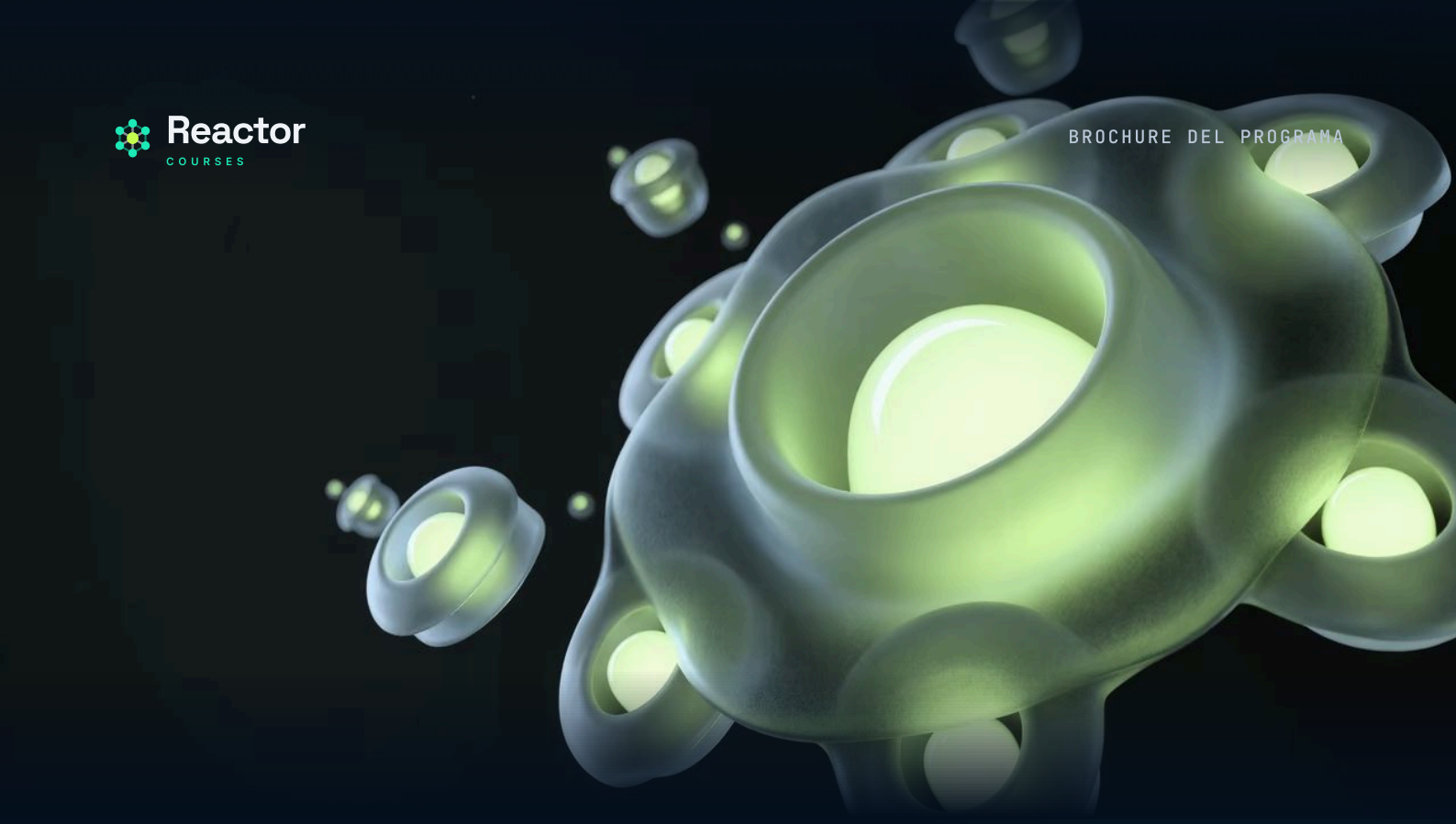
Reactor
COURSES

BROCHURE DEL PROGRAMA



ARQUITECTURA E INGENIERÍA · INTELIGENCIA ARTIFICIAL

Arquitectura de plataforma MLOps

Del experimento al modelo que se reentrena solo

Construye la plataforma que lleva los modelos que entrena tu empresa del experimento a producción y de vuelta: reproducibles, registrados, servidos, monitoreados y reentrenados con un criterio que se puede defender.

Lo que vas a lograr Aprendes a llevar los modelos de machine learning que entrena tu empresa del experimento a producción, y a detectar cuándo hay que reentrenarlos.

- Reproducibilidad
- MLflow y registro
- Pipelines y pruebas
- Feature store
- Servicio de modelos
- Deriva y reentrenamiento

horas en vivo	sesiones	semanas	comando para reproducir toda la plataforma
72	24	12	1

INICIO
Por confirmar

PRECIO
US$400 pago único · USD

MODALIDAD
Clases en vivo por Zoom · Avanzado

reactorcourses.com/programas/mlops-platform-architecture

EL PROGRAMA

Entrenar un modelo es la parte fácil. Que siga funcionando dentro de seis meses es el trabajo.

Construye la plataforma que lleva los modelos que entrena tu empresa del experimento a producción y de vuelta: reproducibles, registrados, servidos, monitoreados y reentrenados con un criterio que se puede defender. Un programa de **72 horas** para ingenieros de ML y de datos que entregan modelos que nadie sabe reproducir, y para equipos de plataforma que deben dar servicio a científicos de datos.

Duración	72 horas · 24 sesiones	Calendario	12 semanas · 2 sesiones por semana de 3 h
Modalidad	Clases en vivo por Zoom	Grabaciones	Disponibles por 12 meses
Credenciales	3 certificados con insignia de Reactor Courses	Nivel	Avanzado
Inicio	Por confirmar	Horario	Miércoles y viernes, 19:00–22:00 (GMT-5)
Precio	US$400 · pago único		

La mayoría de los modelos que entrena una empresa mueren en el notebook o se degradan en producción sin que nadie se entere. Este programa construye la plataforma que evita las dos cosas: cada modelo se puede reproducir, se registra con su linaje, se promueve con criterios escritos, se sirve con su versión, se monitorea contra la realidad y se reentrena cuando hace falta, no cuando alguien se acuerda.

01

Reproducir y promover

Datos versionados, experimentos, registro y documentación

Datos versionados, entornos fijados, experimentos que se encuentran otra vez, un registro con campeón y retador, criterios de promoción escritos como código y una ficha de modelo que responde preguntas de auditoría. La reproducibilidad se valida reproduciendo el modelo de otra persona.

- Versionado de datos con DVC y entornos reproducibles
- Seguimiento de experimentos y registro de modelos con MLflow
- Criterios de promoción por segmento, no solo por promedio
- Ficha de modelo y linaje generados por la plataforma

02

Servir, vigilar y reentrenar

Pipelines, atributos, servicio, deriva y reentrenamiento

Pipelines orquestados y probados, atributos consistentes entre entrenamiento y servicio, la fuga temporal que infla métricas, servicio en lote y en línea, despliegue en sombra, monitoreo de deriva y de desempeño con etiquetas tardías, y un reentrenamiento que se dispara por un umbral justificado en costo.

- Pipeline orquestado y batería de pruebas de ML
- Feature store y consultas en un punto del tiempo
- Empaquetado, servicio y despliegue en sombra
- Deriva, desempeño y reentrenamiento automático con criterio

> Un modelo que no puedes reproducir no es un activo de la empresa: es un recuerdo de la persona que lo entrenó.

PARA QUIÉN ES

¿Es el programa indicado para ti?

Está pensado para quienes ya entrenaron un modelo y quieren llevarlo a producción y mantenerlo funcionando.

Es para ti si

- ✓ Ingenieros de ML o de datos que entregan modelos que nadie sabe reproducir.
- ✓ Ingenieros de plataforma que deben dar servicio a equipos de ciencia de datos.
- ✓ Científicos de datos que quieren llevar sus modelos a producción sin depender de otro equipo.
- ✓ Líderes técnicos que tienen que responder una auditoría sobre un modelo en producción.
- ✓ Equipos con modelos en producción que se degradan sin que nadie lo detecte.

Te conviene otro programa

- – Quien busque aprender ciencia de datos o algoritmos de aprendizaje automático: se dan por sabidos.
- – Quien trabaje solo con modelos de lenguaje por API: aquí el modelo lo entrena la empresa.
- – Quien busque entrenamiento distribuido en GPU: todo el programa corre en CPU.
- – Quien espere un curso de una herramienta: se usan herramientas libres, y el criterio sirve con cualquier otra.
- – Quien no tenga Python con soltura.
- – Quien no pueda dedicar entre 10 y 12 horas por semana durante 12 semanas.

EL RESULTADO

Terminas con una plataforma que reproduce, sirve, vigila y reentrena un modelo, levantada con un comando.

Cada laboratorio deja una pieza de la plataforma del caso, sobre datos abiertos de demanda de bicicletas compartidas. Juntas forman el proyecto que defiendes mientras otra persona la reproduce desde cero.

01 **Datos versionados** Datos crudos, procesados y particionados versionados, con vuelta exacta a una versión anterior.

02 **Modelo reproducido por otro** El modelo de otro alumno reproducido con su repositorio y un comando, dentro de la tolerancia declarada.

03 **Experimentos y registro** Veinte experimentos registrados con versión de datos y código, y un registro con campeón y retador.

04 **Promoción con criterio** Criterios como código que bloquean un modelo mejor en promedio pero peor en un segmento.

05 **Ficha de modelo** Ficha generada por la plataforma que responde preguntas de auditoría por sí sola.

06 **Pipeline y pruebas** Pipeline orquestado que reanuda desde el paso fallido, y pruebas que detectan datos y modelos defectuosos.

07 **Atributos sin desfase** Feature store que devuelve el mismo valor al entrenar y al predecir, y una fuga temporal detectada y corregida.

08 **Servicio en dos modos** Modelo empaquetado servido en lote y en línea, con latencia, costo y versión por predicción.

09 **Despliegue en sombra** Retador evaluado en sombra contra el campeón y decisión de promoción con vuelta atrás.

10 **Deriva y desempeño** Monitoreo que no alerta por la estacionalidad normal y sí por una deriva real, más desempeño con etiquetas tardías.

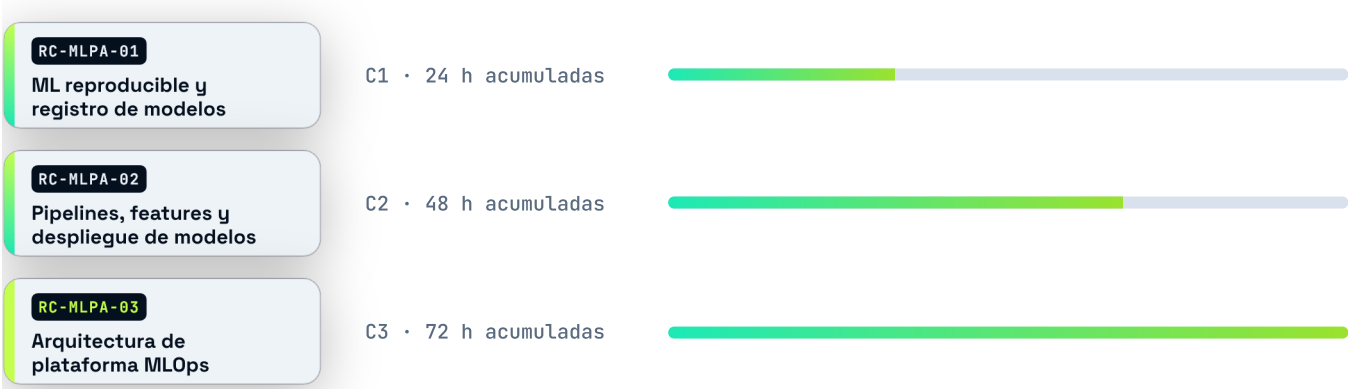
11 **Reentrenamiento automático** Reentrenamiento disparado por un umbral justificado en costo, que pasa los criterios de promoción sin intervención.

12 **Auditoría respondida** Quién entrenó, con qué datos, quién aprobó y qué versión respondió, contestado solo con la plataforma.

RUTA DE CERTIFICACIÓN

3 credenciales progresivas, no una al final.

Cada etapa cierra un bloque completo: primero reproducir y promover, después pipelines, atributos y servicio, y al final monitoreo, reentrenamiento y plataforma. Son credenciales de Reactor Courses, no certificaciones de terceros.



CERTIFICACIÓN INTEGRAL

MLOps Platform Architecture — Del experimento al modelo que se reentrena solo (72 horas)

Requiere entregar la plataforma reproducible desde un comando con el episodio de deriva y reentrenamiento demostrado. Los certificados son emitidos por Reactor Courses y acreditan las horas y competencias del programa.

Cada certificado de Reactor Courses viene con su insignia, aparte de cualquier certificación oficial.

Cualquiera puede comprobar un certificado de Reactor Courses con su código en reactorcourses.com/verificar.

TEMARIO

Las 24 sesiones, abiertas y sin letra chica.

72 horas repartidas en 8 módulos. Así sabes qué verás, qué practicarás y qué entregarás en cada una de las 12 semanas.

MÓDULO 1 EL CICLO DE VIDA DE UN MODELO PROPIO — 6 h · 2 sesiones

01 Del notebook a producción, y de vuelta

ciclo de vida de un modelo entrenado por la empresa · diferencias con consumir un modelo fundacional por API · deuda técnica oculta de los sistemas de aprendizaje automático · el caso: predicción de demanda de bicicletas compartidas con datos abiertos · licencia del conjunto de datos y su atribución · el entorno local: contenedores, servidor de seguimiento y almacén de artefactos

LABORATORIO: Levantar el entorno del caso, entrenar el modelo base desde el notebook entregado y documentar las siete cosas que harían falta para que otra persona lo reproduzca y lo ponga en producción

02 Madurez y prioridades

niveles de madurez de MLOps según la literatura pública de proveedores · qué se automatiza primero y qué nunca · roles: ciencia de datos, ingeniería de ML y plataforma · costo de la falta de reproducibilidad en una auditoría · señales de que un equipo necesita plataforma y no más modelos · plan de cien días

LABORATORIO: Diagnosticar la madurez de MLOps de una organización ficticia a partir de entrevistas y repositorios entregados, y priorizar las cinco automatizaciones de mayor retorno con su justificación

MÓDULO 2 REPRODUCIBILIDAD — 9 h · 3 sesiones

03 Datos versionados

versionar datos sin meterlos en Git · DVC: seguimiento, almacenamiento remoto y etapas · instantáneas de datos frente a datos que cambian en el origen · conjuntos de entrenamiento, validación y prueba congelados · linaje del dato al modelo · volumen y costo del almacenamiento de versiones

LABORATORIO: Versionar con DVC los datos crudos, procesados y particionados del caso, y demostrar que se puede volver a la versión de hace tres iteraciones y obtener exactamente el mismo conjunto de entrenamiento

04 Entorno, código y semilla

fuentes de no determinismo en el entrenamiento · dependencias fijadas y entorno en contenedor · semillas y su alcance real · parámetros como configuración, no como código · un comando que reproduce todo · tolerancia aceptable entre dos ejecuciones

LABORATORIO: Reproducir el modelo de otro alumno solo con su repositorio, su imagen de contenedor y un comando, comparar métricas y predicciones contra las suyas dentro de la tolerancia declarada y reportar cada fuente de diferencia encontrada

05 Seguimiento de experimentos

MLflow: ejecuciones, parámetros, métricas y artefactos · comparar experimentos sin una planilla · registrar la versión de datos y de código en cada ejecución · búsqueda de hiperparámetros con seguimiento · experimentos que nadie puede volver a encontrar · convenciones de nombres y etiquetas

LABORATORIO: Registrar en MLflow veinte experimentos del caso con versión de datos y de código, y elegir el ganador con una consulta reproducible que cualquier otro alumno pueda volver a ejecutar

MÓDULO 3 REGISTRO, PROMOCIÓN Y DOCUMENTACIÓN

9 h · 3 sesiones

06 Registro de modelos

registro de modelos de MLflow: versiones y alias · firma del modelo: entradas y salidas esperadas · dependencias empaquetadas con el modelo · modelo campeón y modelo retador · quién puede registrar y quién puede promover · retirar una versión sin romper a quien la usa

LABORATORIO: Registrar las versiones del modelo con firma y dependencias, asignar los alias de campeón y retador, y demostrar que el servicio carga siempre el campeón aunque se registren versiones nuevas

07 Criterios de promoción

métrica de negocio frente a métrica del modelo · umbral mínimo y comparación contra el campeón · evaluación por segmentos: el promedio que esconde un grupo · validación de equidad básica · aprobación humana en la promoción · registro de por qué se promovió

LABORATORIO: Escribir los criterios de promoción del caso como código —métrica, segmentos y comparación contra el campeón— y demostrar que bloquean un modelo con mejor promedio pero peor en los días de lluvia

08 Ficha de modelo y linaje

ficha de modelo según Model Cards for Model Reporting · uso previsto y usos no previstos · datos, métricas por segmento y limitaciones conocidas · linaje del dato a la predicción · aprobaciones y su registro · la ficha generada desde la plataforma y no a mano

LABORATORIO: Generar automáticamente desde MLflow la ficha del modelo campeón con datos, métricas por segmento, limitaciones y linaje, y verificar con otro alumno que puede responder cinco preguntas de auditoría solo con la ficha

MÓDULO 4 PIPELINES, PRUEBAS Y ATRIBUTOS

12 h · 4 sesiones

09 Pipeline de entrenamiento orquestado

pasos del pipeline: datos, validación, entrenamiento, evaluación y registro · Prefect como orquestador del laboratorio · reintentos, caché de pasos y ejecución parcial · parámetros y disparadores del pipeline · el pipeline como código revisable · cuándo un orquestador de datos alcanza y cuándo no

LABORATORIO: Convertir el entrenamiento del caso en un pipeline de Prefect con validación, evaluación y registro, y demostrar que una falla en la evaluación reanuda desde ese paso sin repetir el procesamiento de datos

10 Pruebas de aprendizaje automático

pruebas de esquema y de rangos de datos con pandera · pruebas de invariancia y de dirección esperada del modelo · rendimiento mínimo como prueba · pruebas del código de atributos · pruebas de la infraestructura de servicio · qué prueba corre en cada etapa de la tubería

LABORATORIO: Escribir la batería de pruebas de datos, modelo e infraestructura del caso e integrarla a la tubería, demostrando que detecta un cambio de unidades en la temperatura y un modelo que predice menos demanda con mejor clima

11 Atributos consistentes con un feature store

desfase entre entrenamiento y servicio · feature store: concepto, almacén en línea y fuera de línea · Feast: definición de atributos, materialización y consulta · reutilizar atributos entre modelos · frescura del atributo y su efecto en la predicción · cuándo un feature store es exceso de ingeniería

LABORATORIO: Definir en Feast los atributos del caso, servirlos fuera de línea al entrenamiento y en línea a la predicción, y demostrar que ambas rutas producen el mismo valor para la misma estación y hora

12 Fuga temporal y consultas en un punto del tiempo

fuga de información desde el futuro · uniones en un punto del tiempo · validación temporal en lugar de partición aleatoria · etiquetas que se conocen tarde · el modelo que era excelente hasta producción · detectar la fuga con pruebas

LABORATORIO: Introducir a propósito una fuga temporal en el caso, demostrar que infla la métrica de validación, y corregirla con uniones en un punto del tiempo y validación temporal, midiendo la métrica real

MÓDULO 5 SERVICIO

12 h · 4 sesiones

13 Empaquetar el modelo

el modelo como artefacto inmutable · BentoML: servicio, dependencias e imagen · contrato de entrada y salida del servicio · validación de la entrada antes de predecir · versión del modelo en cada respuesta · imagen reproducible desde el registro

LABORATORIO: Empaquetar el modelo campeón con BentoML como imagen construida desde el registro, con validación de entrada y versión en cada respuesta, y demostrar que una petición mal formada se rechaza con un error útil

14 Predicción en lote y en línea

predicción en lote programada · predicción en línea con latencia acotada · precalcular frente a calcular al pedir · atributos disponibles en cada modo · costo de cada modo según el volumen · registro de predicciones para monitorear después

LABORATORIO: Servir el mismo modelo en lote para el plan de redistribución nocturno y en línea para la consulta por estación, comparar latencia y costo, y registrar cada predicción con su versión para el monitoreo

15 Despliegue en sombra y modelo retador

despliegue en sombra: predecir sin afectar al usuario · comparar campeón y retador con tráfico real · canario de modelos y su métrica de corte · pruebas A/B y el tiempo que necesitan · volver al campeón en segundos · cuando el retador gana en métricas y pierde en negocio

LABORATORIO: Poner el modelo retador en sombra junto al campeón, comparar sus predicciones sobre una semana de tráfico simulado y decidir la promoción con los criterios de la sesión 07, revirtiendo al campeón si el retador empeora un segmento

16 Latencia, capacidad y costo de inferencia

latencia de inferencia en CPU y sus componentes · lotes dinámicos y concurrencia · formato del modelo y ONNX como alternativa · capacidad necesaria para el pico · costo por mil predicciones · cuándo hace falta una GPU y cuándo no

LABORATORIO: Medir latencia y capacidad del servicio con y sin lotes dinámicos y con el modelo exportado a ONNX, y calcular el costo por mil predicciones de cada variante para el pico de demanda del caso

MÓDULO 6 MONITOREO Y REENTRENAMIENTO

12 h · 4 sesiones

17 Deriva de datos y de predicciones

deriva de datos, de concepto y de predicciones · pruebas estadísticas de deriva y su sensibilidad · Evidently para informes y monitoreo · ventana de referencia y ventana actual · deriva que no importa para el negocio · alerta de deriva sin falsos positivos diarios

LABORATORIO: Monitorear con Evidently la deriva del caso entre 2011 y 2012, ajustar el umbral para que no alerte por la estacionalidad normal y demostrar que sí alerta ante un cambio inducido en la distribución del clima

18 Desempeño con etiquetas tardías

cuando la etiqueta llega horas o meses después · unir predicción y resultado real · métricas aproximadas mientras llega la etiqueta · desempeño por segmento en producción · el modelo que se degrada sin deriva visible · tablero de salud del modelo

LABORATORIO: Unir las predicciones registradas con la demanda real que llega al día siguiente, calcular el desempeño diario por segmento y demostrar que el tablero detecta una degradación que el monitoreo de deriva no vio

19 Reentrenamiento con criterio

reentrenar por calendario, por deriva o por desempeño · umbral justificado con costo de error y costo de cómputo · reentrenamiento que pasa por los mismos criterios de promoción · datos nuevos y datos viejos: qué ventana usar · evitar reentrenar sobre datos contaminados · registro de cada reentrenamiento

LABORATORIO: Implementar el disparo de reentrenamiento por desempeño con un umbral justificado en costo, y demostrar que tras una deriva inducida el pipeline reentrena, el nuevo modelo pasa los criterios de promoción y reemplaza al campeón sin intervención

20 **Gobierno del ciclo de vida**

aprobaciones y separación de funciones · auditoría: quién entrenó, aprobó y desplegó qué · retiro de un modelo y de sus atributos · inventario de modelos en producción · requisitos regulatorios que llegan como preguntas de auditoría · modelos huérfanos sin dueño

LABORATORIO: Responder una auditoría simulada sobre el modelo en producción —quién lo entrenó, con qué datos, quién lo aprobó y qué versión respondió una predicción concreta— usando solo la plataforma, y retirar una versión vieja con sus atributos

MÓDULO 7 LA PLATAFORMA COMO SERVICIO

6 h · 2 sesiones

21 **Autoservicio para equipos de ciencia de datos**

plantilla de proyecto de aprendizaje automático · lo que la plataforma da y lo que decide cada equipo · límites de recursos por equipo · documentación para que un equipo nuevo empiece solo · métrica de la plataforma: tiempo del notebook a producción · adopción y resistencia

LABORATORIO: Crear la plantilla de proyecto de la plataforma y medir cuánto tarda otro alumno en llevar un modelo nuevo de su notebook al registro y al servicio usando solo la plantilla y la documentación

22 **Costo de cómputo del ciclo de vida**

costo de entrenar, de evaluar y de servir · experimentos que nadie apaga · caché de pasos y de atributos como ahorro · costo por reentrenamiento y por mil predicciones · nube frente a local para cada etapa · informe de costo para el negocio

LABORATORIO: Calcular el costo mensual del ciclo de vida del caso en local y en una nube de referencia con precios de lista, identificar las dos etapas más caras y reducir una de ellas midiendo el ahorro

MÓDULO 8 PROYECTO INTEGRADOR

6 h · 2 sesiones

23 **Deriva inducida y reentrenamiento de punta a punta**

todo el ciclo desde un comando · deriva inducida a propósito · detección, reentrenamiento y promoción automáticos · verificación del modelo nuevo en producción · trazabilidad de todo el episodio · registro para la defensa

LABORATORIO: Reproducir la plataforma desde un comando en un entorno limpio, inducir una deriva y demostrar que la detección, el reentrenamiento, la promoción y el despliegue ocurren solos y quedan trazados

24 **Defensa de la plataforma**

reproducibilidad demostrada · criterios de promoción y gobierno · servicio, monitoreo y reentrenamiento · costo del ciclo de vida · decisiones con sus alternativas · preguntas cruzadas del grupo

LABORATORIO: Defender ante el grupo la plataforma del caso mientras otro alumno la reproduce desde cero, respondiendo con datos de la propia plataforma sobre reproducibilidad, promoción, deriva y costo

LO QUE VAS A USAR

Software libre y mantenido, elegido por la etapa del ciclo de vida que resuelve.

Reproducibilidad

DVC · Contenedores

Experimentos y registro

MLflow

Pipelines y pruebas

Prefect · pandera

Atributos

Feast

Servicio

BentoML ONNX

Monitoreo

Evidently

Los logos son marcas de sus respectivos dueños. Se muestran solo para identificar las herramientas que se usan en el programa y no implican aval, patrocinio ni afiliación.

CÓMO FUNCIONA

Clases en vivo, un modelo real y una plataforma que otra persona tiene que poder reproducir.

01 **Clases en vivo por Zoom** 24 sesiones de 3 horas, 2 veces por semana. Se construye en vivo y se revisa cada pieza con el grupo.

02 **Todo queda grabado** Si te pierdes una sesión o vives en otro huso horario, la grabación queda disponible. Acceso por 12 meses.

03 **30% concepto / 70% laboratorio** Cada concepto se aplica en el mismo día a la plataforma del caso, con un criterio de aceptación verificable.

04 **Reproducir lo de otro** La reproducibilidad no se declara: se valida reproduciendo el modelo de otro alumno solo con su repositorio.

05 **Todo en CPU y en local** Software libre en contenedores sobre tu máquina. Sin GPU ni cuenta de nube obligatoria.

06 **Datos abiertos con licencia** El caso usa un conjunto de datos público con licencia de atribución, citado como corresponde.

Horarios · Primer grupo

Miércoles	19:00 - 22:00
Viernes	19:00 - 22:00

Horario referido a GMT-5 (Perú, Colombia, Ecuador y Panamá; en Ciudad de México es una hora menos).

Los días y el horario son la propuesta para este grupo y se confirman junto con la fecha de inicio, antes de abrir la matrícula. El calendario definitivo se entrega al confirmar la matrícula.

REQUISITOS

Lo que necesitas traer a la sesión 01.

No hay nivelación previa. La sesión 01 parte de un notebook que funciona y pregunta qué falta para que alguien más lo use.

Aprendizaje automático

- ✓ Haber entrenado y evaluado un modelo
- ✓ scikit-learn o equivalente
- ✓ Métricas de evaluación y validación

Programación

- ✓ Python con soltura
- ✓ pandas
- ✓ Git

Datos

- ✓ SQL
- ✓ Nociones de series de tiempo

Herramientas

- ✓ Docker a nivel de uso
- ✓ Nociones de una nube pública

Equipo y tiempo

- ✓ Computadora con 16 GB de memoria
- ✓ Seis horas de clase por semana
- ✓ Entre cuatro y seis horas de laboratorio
- ✓ 12 semanas de compromiso

Costos de herramientas que asumes tú `US$0 a 30 aprox.`

Estos costos NO están incluidos en el precio del programa. Todo el software del laboratorio es gratuito y el conjunto de datos del caso es abierto.

- El laboratorio corre en tu máquina con contenedores y software libre; no hace falta GPU
- Si tu equipo no alcanza (se recomiendan 16 GB de memoria), una máquina virtual alquilada solo durante las prácticas cuesta del orden de US$20 en todo el programa
- La comparación de costo con una nube se hace con precios de lista, sin desplegar nada

PRECIO

72 horas para que tus modelos sobrevivan a quien los entrenó.

US$400

Pago único · USD · Incluye impuestos

Todavía no hay fecha de inicio. Escríbenos y te avisamos cuando la confirmemos: no cobramos nada antes de abrir la matrícula.

Incluye

- ✓ 24 sesiones en vivo por Zoom (72 horas lectivas)
- ✓ Grabación de todas las clases con acceso por 12 meses
- ✓ Entorno de la plataforma listo para levantar con un comando
- ✓ Notebook de partida del caso y datos abiertos preparados
- ✓ Plantilla de proyecto de aprendizaje automático y de ficha de modelo
- ✓ Los 3 certificados progresivos
- ✓ Canal privado del grupo para dudas entre sesiones
- ✓ Actualizaciones del material por 12 meses

No incluye

- — Una máquina virtual alquilada si tu equipo no alcanza (rango de costos propios: US$0 a 30)
- — Exámenes ni certificaciones de terceros: este programa no prepara para ninguno
- — Plataformas comerciales de aprendizaje automático: todo el laboratorio usa software libre

Métodos de pago

Tarjeta de crédito · Tarjeta de débito internacional · PayPal · Apple Pay · Google Pay

Garantía de 14 días. Si dentro de los primeros 14 días del programa decides que no es para ti, te devolvemos el 100 % de lo pagado. No pedimos justificación.

Paddle actúa como comerciante registrado (merchant of record): emite el comprobante y gestiona los impuestos de cada país.

PREGUNTAS FRECUENTES

Preguntas frecuentes sobre el programa.

¿Las clases son en vivo o grabadas?

En vivo por Zoom: 24 sesiones de 3 horas, 2 veces por semana. Todas quedan grabadas y disponibles por 12 meses.

¿Cuándo empiezan las clases?

La fecha de inicio y el calendario definitivo todavía no están confirmados: se publican en esta misma ficha antes de que se abra la matrícula. Los días propuestos son Miércoles y viernes en horario de GMT-5 (Perú, Colombia, Ecuador y Panamá; en Ciudad de México es una hora menos). Si quieres que te avisemos cuando se confirmen, escribe a soporte@reactorcourses.com.

¿Qué nivel necesito?

Haber entrenado y evaluado un modelo con scikit-learn o equivalente, Python con soltura y SQL. No hace falta experiencia previa en producción: eso es lo que enseña el programa.

¿Se ven modelos de lenguaje o IA generativa?

No. El programa trata modelos que la empresa entrena con sus datos. Servir modelos fundacionales a toda una organización es otro problema, con otras herramientas, y no se cubre aquí.

¿Necesito GPU o cuenta de nube?

No. Todo corre en CPU, en tu máquina, con contenedores. Se recomienda un equipo con 16 GB de memoria; si no alcanza, el rango de costos propios es de US$0 a 30, por alquilar una máquina virtual solo durante las prácticas.

¿Se cruza con otros programas del catálogo?

Poco. Cerca de una de cada seis sesiones toca temas que un programa de plataforma de datos o de plataforma de IA también trata: orquestación de pipelines, fuga temporal, deriva y plantillas de autoservicio. Aquí se aplican al ciclo de vida de un modelo entrenado, con otros laboratorios. No hay descuento ni convalidación entre programas.

¿Por qué estas herramientas y no una plataforma comercial?

Porque son libres, se mantienen activamente y cubren cada etapa del ciclo. Se descartaron herramientas de servicio de modelos con licencias que dejaron de ser de código abierto. Lo que se aprende se traslada a cualquier plataforma gestionada de las nubes.

¿Qué datos se usan?

Un conjunto de datos público de demanda de bicicletas compartidas, con licencia de atribución, que trae una deriva natural entre sus dos años. Se cita como corresponde en la ficha del modelo.

¿Qué pasa si falto a una clase?

La grabación queda disponible y cada laboratorio tiene su punto de partida en el repositorio. La plataforma se construye por capas: faltar a varias sesiones seguidas sin recuperarlas te deja sin piezas del proyecto final.

¿Los certificados tienen validez oficial?

No. Son certificados de Reactor Courses que acreditan las horas y competencias del programa. No son títulos universitarios ni certificaciones de terceros.

¿Cuánto tiempo debo dedicarle por semana?

Seis horas de clase más entre cuatro y seis horas de laboratorio. Considera entre 10 y 12 horas semanales durante 12 semanas.

¿Qué métodos de pago aceptan?

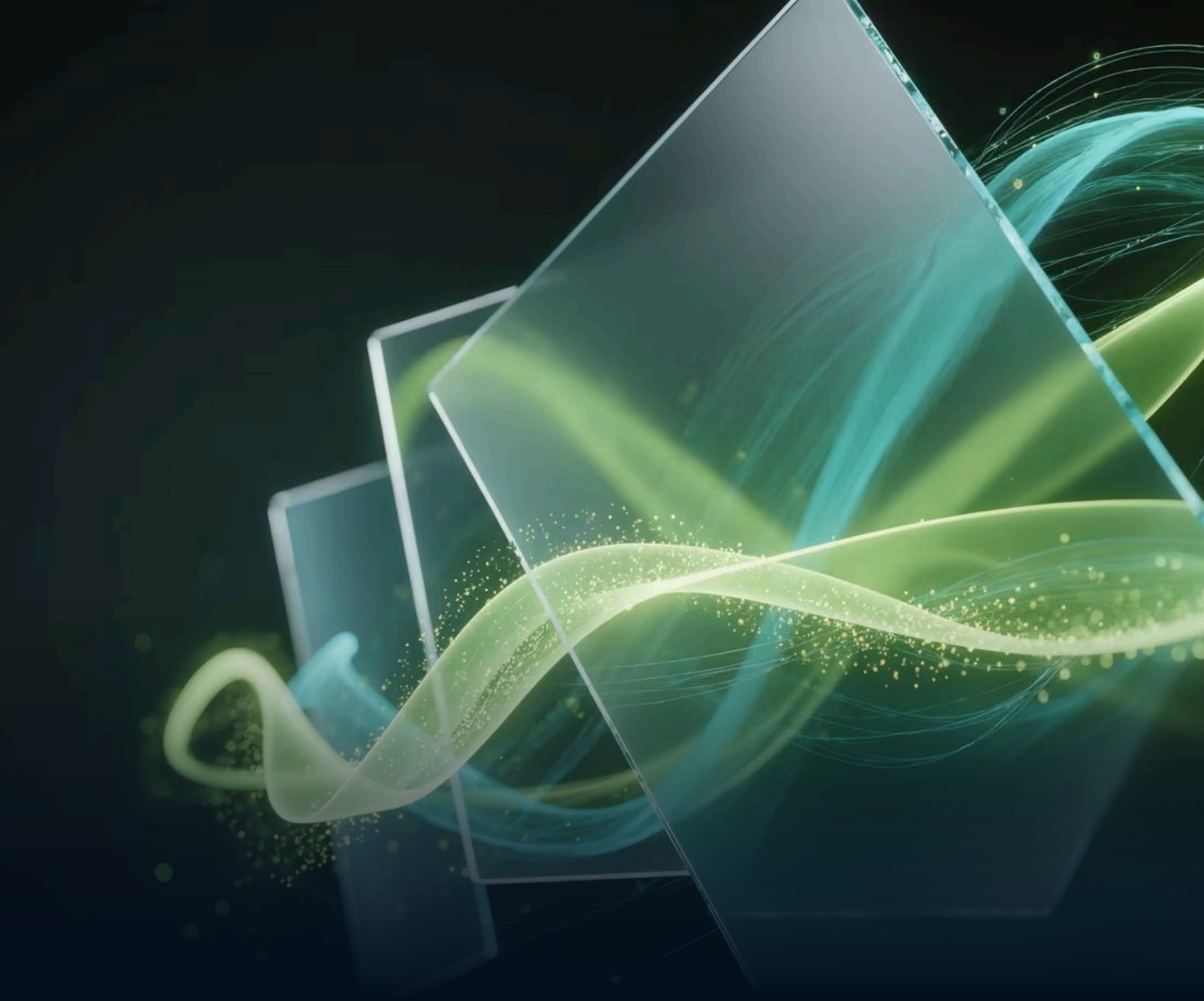
El pago lo procesa Paddle y acepta Tarjeta de crédito, Tarjeta de débito internacional, PayPal, Apple Pay, Google Pay. El cobro es en dólares. Importante: no procesamos Yape, Plin, OXXO, Mercado Pago ni transferencias locales. Si tu banco rechaza compras internacionales, habilítalas antes de pagar o usa PayPal.

¿Emiten factura o comprobante?

Sí. Paddle actúa como comerciante registrado (merchant of record): emite el comprobante y gestiona los impuestos de cada país. Lo recibirás por correo automáticamente al completar el pago.

¿Cuál es la política de reembolso?

Tienes 14 días desde el inicio del programa para pedir el reembolso completo escribiendo a soporte@reactorcourses.com. Publicamos el temario completo y los requisitos justamente para que decidas informado.



Reactor
COURSES

Ingeniería de software en la era de los agentes

¿Te quedó alguna duda?

Escríbenos a info@reactorcourses.com. La ficha al día, con el temario interactivo y la inscripción, está en reactorcourses.com/programas/mlops-platform-architecture.

`reactorcourses.com/programas/mlops-platform-architecture`

Información vigente al 23 de noviembre de 2026. El precio, las fechas y el temario pueden cambiar: la versión al día es la de la web.

Reactor Courses no está afiliado a Microsoft, Amazon Web Services, OpenAI, Anthropic ni Google. Los nombres de productos se mencionan únicamente con fines descriptivos del contenido académico.