Reactor
COURSES

BROCHURE DEL PROGRAMA



PREPARACIÓN PARA CERTIFICACIONES · CERTIFICACIONES DE MICROSOFT AZURE

AI-200 · Azure AI Cloud Developer Associate

Preparación para la certificación

Construye en Azure lo que sostiene a una aplicación de IA —contenedores, datos con búsqueda vectorial, mensajería, secretos y trazas— y llega al examen AI-200 habiéndolo desplegado y diagnosticado con tus manos.

Lo que vas a lograr Preparación para el examen AI-200: desplegar en Azure los contenedores, las bases de datos, la mensajería y la seguridad que usa una aplicación con IA.

Certificación: te preparamos para aprobar AI-200, que emite Microsoft. Además te llevas los certificados e insignias de Reactor Courses.

- Contenedores
- Búsqueda vectorial
- Mensajería y eventos
- Azure Functions
- Seguridad y observabilidad
- Simulacro de examen

horas en vivo	sesiones	semanas	dominios del examen
60	20	10	4

INICIO
Por confirmar

PRECIO
US$150 pago único · USD

MODALIDAD
Clases en vivo por Zoom · Intermedio

reactorcourses.com/programas/ai-200-azure-ai-cloud-developer

EL PROGRAMA

Un examen sobre la nube que sostiene a una aplicación de IA, no sobre el modelo.

Construye en Azure lo que sostiene a una aplicación de IA —contenedores, datos con búsqueda vectorial, mensajería, secretos y trazas— y llega al examen AI-200 habiéndolo desplegado y diagnosticado con tus manos. Un programa de **60 horas** para desarrolladores que construyen servicios en Azure y quieren llegar al examen habiendo desplegado, conectado y diagnosticado cada pieza, no solo leído sobre ella.

Duración	60 horas · 20 sesiones	Calendario	10 semanas · 2 sesiones por semana de 3 h
Modalidad	Clases en vivo por Zoom	Grabaciones	Disponibles por 12 meses
Examen oficial	AI-200 · lo emite Microsoft	Credenciales	3 certificados con insignia de Reactor Courses
Nivel	Intermedio	Inicio	Por confirmar
Horario	Martes y jueves, 19:00–22:00 (GMT-5)	Precio	US$150 · pago único

El AI-200 evalúa si sabes construir en Azure lo que rodea a una aplicación con IA: los contenedores donde corre, las bases de datos donde busca por similitud, la mensajería que conecta sus servicios y la seguridad y las trazas que permiten operarla. Este programa recorre los cuatro dominios del examen construyendo un mismo servicio, sesión a sesión.

01

Los cuatro dominios, con las sesiones según su peso

Cobertura completa del temario publicado

El programa reparte las sesiones según el peso de cada dominio: cinco para datos con búsqueda vectorial, que es el de mayor peso, y cuatro para cada uno de los otros tres. Cada objetivo del temario publicado por Microsoft cae en al menos una sesión.

- Contenedores: Container Registry, App Service, Container Apps y AKS
- Datos: Cosmos DB, PostgreSQL con pgvector y Azure Managed Redis, con búsqueda vectorial
- Conectar servicios: Service Bus, Event Grid y Azure Functions
- Seguridad y monitoreo: Key Vault, App Configuration, OpenTelemetry y KQL

02

Un solo caso, construido sesión a sesión

Laboratorio en cada sesión sobre el servicio de posventa de una distribuidora de repuestos

Todos los laboratorios trabajan sobre el mismo caso: un catálogo de piezas que el cliente describe con sus propias palabras, un historial de tickets, pedidos que viajan entre servicios y un inventario que avisa cuando falta stock. Cada sesión agrega una pieza y la última se defiende desplegada, sin claves y con su traza de punta a punta.

- Un laboratorio con criterio de aceptación verificable en cada sesión de contenido
- Fallas provocadas a propósito para practicar el diagnóstico
- Costo medido y recursos destruidos al terminar cada laboratorio
- Una sesión de preguntas de caso con simulacro cronometrado y una de defensa del proyecto

> El modelo es una llamada a una API. Lo que hay que saber construir, asegurar y diagnosticar es todo lo que la rodea.

PARA QUIÉN ES

¿Es el programa indicado para ti?

Está pensado para desarrolladores que ya programan y quieren certificarse construyendo en Azure la infraestructura de aplicaciones con IA. Partimos de lo que ya sabes y vamos directo a lo que pide el examen.

Es para ti si

- ✓ Desarrolladores que construyen servicios en Azure y quieren certificarse como desarrolladores de la nube de Microsoft.
- ✓ Desarrolladores que venían preparando el AZ-204, ya retirado, y buscan el examen de desarrollador de Azure vigente.
- ✓ Equipos que agregan búsqueda vectorial a sus aplicaciones y tienen que decidir entre Cosmos DB, PostgreSQL y Redis.
- ✓ Desarrolladores backend que tienen que pasar sus servicios a contenedores y conectarlos con colas y eventos en Azure.
- ✓ Quien tiene una aplicación en Azure que falla sin que nadie sepa por qué y quiere aprender a trazarla y diagnosticarla.

Te conviene otro programa

- – Quien no programe: todos los laboratorios se escriben en Python y se espera soltura con el lenguaje desde la sesión 01.
- – Quien nunca haya usado Azure ni contenedores: se asume que sabes crear recursos, asignar roles y construir una imagen de Docker.
- – Quien quiera aprender a construir agentes o a trabajar con modelos de IA: este examen evalúa la nube alrededor del modelo, no el modelo. Para eso la escuela tiene otro programa de preparación.
- – Quien administra infraestructura y no escribe código: el examen pregunta cómo se programa contra cada servicio, no solo cómo se configura.
- – Quien busque el certificado rápido: aquí se construye un laboratorio por sesión y eso lleva tiempo fuera de clase.
- – Quien no pueda pagar el consumo de su suscripción de Azure y la tasa del examen, ni dedicar entre 10 y 12 horas por semana durante 10 semanas.

EL RESULTADO

Terminas con un servicio de posventa desplegado en Azure, sin claves y trazado de punta a punta.

Cada laboratorio deja un artefacto con su criterio de aceptación cumplido. Juntos forman el proyecto integrador que defiendes en la última sesión.

01 **Entorno sin claves** Entorno base desplegado con Bicep, con presupuesto y alertas, y una aplicación que se autentica igual en local y en Azure sin ninguna clave.

02 **Imágenes que se reconstruyen solas** Registro de contenedores con una tarea que recompila la API al hacer commit y al cambiar la imagen base.

03 **Tres formas de correr un contenedor** La misma API desplegada en App Service con ranuras, en Container Apps con revisiones y escalado, y en AKS con manifiestos, con la comparación escrita.

04 **Fallas diagnosticadas** Revisiones y pods que no arrancan, con la causa encontrada en los registros y la corrección documentada.

05 **Cosmos DB medido en RU** Historial de tickets con la política de indexación ajustada, el costo de cada consulta medido y búsqueda vectorial que se mantiene al día con el change feed.

06 **Búsqueda semántica con filtros** Catálogo en PostgreSQL con pgvector que encuentra la pieza aunque el cliente la describa con otras palabras, con sus aciertos medidos.

07 **Caché que se invalida a tiempo** Caché en Azure Managed Redis con invalidación por cambio de precio y caché semántica, con la tasa de aciertos y la latencia medidas.

08 **Pedidos que no se pierden** Colas y temas de Service Bus con enrutamiento por filtros, mensajes fallidos reprocesables y un trabajador que escala según la cola.

09 **Eventos y funciones** Eventos de inventario filtrados con reintentos y una API serverless en Azure Functions conectada por identidad.

10 **Secretos que rotan sin reinicio** Un secreto rotado en Key Vault que la aplicación toma sin reiniciarse, y la configuración centralizada en App Configuration.

11 **Una traza de punta a punta** Un pedido seguido en una sola traza de OpenTelemetry que cruza HTTP y mensajería, y tres fallas resueltas con consultas KQL que quedan como alertas.

CERTIFICACIÓN OFICIAL

Te preparamos para aprobar AI-200.

La certificación oficial es el objetivo del programa. La emite Microsoft cuando apruebas su examen, que se agenda, se paga y se rinde aparte, con Microsoft, en un centro de examen o en línea.

AI-200

Azure AI Cloud Developer

La emite Microsoft · Nivel Associate

Cómo te preparamos

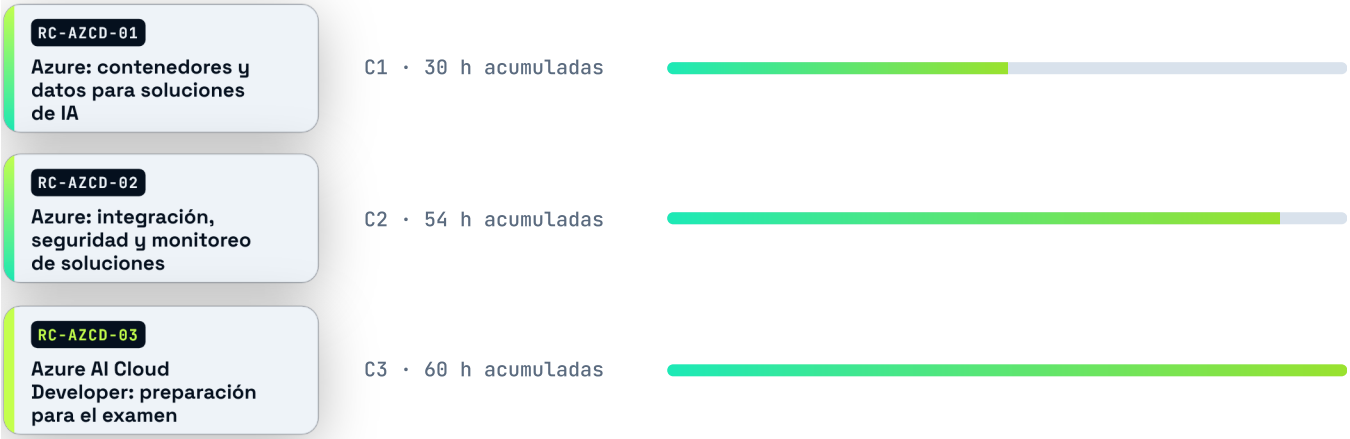
1. **Sobre el temario oficial** Cada sesión cubre los dominios que Microsoft publica para el examen, con el peso que tiene cada uno.
2. **Práctica en cada sesión** Laboratorios propios: lo que pregunta el examen lo haces antes con tus manos.
3. **Simulacro de examen** Practicas con el formato del examen y repasas lo que fallaste antes de agendarlo.

Reactor Courses no emite la certificación oficial ni está avalada por Microsoft.

ADEMÁS: CERTIFICADOS E INSIGNIAS DE REACTOR COURSES

3 credenciales progresivas, no una al final.

Cada etapa cierra un bloque completo del programa. Son certificados de Reactor Courses que acreditan las horas cursadas: la credencial Azure AI Cloud Developer Associate la emite Microsoft y exige aprobar su examen.



CERTIFICACIÓN INTEGRAL

AI-200 · Azure AI Cloud Developer — Preparación para el examen (60 horas)

Requiere entregar los laboratorios de cada bloque, el simulacro y la defensa del proyecto integrador. Los certificados son emitidos por Reactor Courses y acreditan las horas y competencias del programa de preparación.

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 20 sesiones, abiertas y sin letra chica.

60 horas repartidas en 6 módulos. Así sabes qué verás, qué practicarás y qué entregarás en cada una de las 10 semanas.

MÓDULO 1 PUNTO DE PARTIDA: ENTORNO, IDENTIDAD Y MÉTODO DEL EXAMEN — 3 h · 1 sesión

01 Entorno del programa, autenticación sin claves y el método del examen

suscripción, grupo de recursos y convenciones de nombres del programa · CLI de Azure y Bicep para desplegar el entorno base · DefaultAzureCredential: la misma autenticación en la máquina del desarrollador y en Azure · identidad administrada y asignación de roles para una aplicación · presupuesto y alertas de costo en Azure Cost Management · estructura del examen: cuatro dominios, pesos, 120 minutos y puntaje de 700 · el caso de todo el programa: el servicio de posventa de una distribuidora de repuestos industriales

LABORATORIO: Desplegar con Bicep el entorno base del caso con presupuesto y alertas al 50, 75 y 90 %, y una aplicación Python mínima que lee un recurso con DefaultAzureCredential en local y con identidad administrada en Azure, sin ninguna clave en el código ni en la configuración

MÓDULO 2 SOLUCIONES EN CONTENEDORES — dominio 1 del examen (20-25%) — 12 h · 4 sesiones

02 Imágenes, Azure Container Registry y ACR Tasks

Dockerfile de varias etapas para un servicio Python · crear un registro y elegir su nivel · autenticación al registro con identidad administrada y roles de extracción y publicación · compilar en la nube con az acr build · ACR Tasks disparadas por un commit y por la actualización de la imagen base · etiquetas, digest y por qué desplegar por digest · limpieza de imágenes sin etiqueta

LABORATORIO: Publicar la API del catálogo de repuestos en el registro con una ACR Task que la reconstruye al hacer commit y al cambiar la imagen base; demostrar con el historial de ejecuciones que las dos condiciones dispararon la compilación

03 App Service para contenedores: configuración, secretos y despliegue

plan de App Service y aplicación web para contenedores · extraer la imagen del registro con identidad administrada · configuración de la aplicación y variables de entorno · referencias a Key Vault en la configuración de la aplicación · ranuras de despliegue e intercambio · registros del contenedor y transmisión de registros · puerto, comprobación de estado y reinicios del contenedor

LABORATORIO: Desplegar la API en App Service desde el registro sin credenciales del registro, con la cadena de conexión como referencia a Key Vault; promover una versión nueva desde una ranura de staging con intercambio y mostrar que las peticiones no fallaron durante el cambio

04 Azure Container Apps: entornos, revisiones y escalado con KEDA

entorno de Container Apps y su red · aplicación, revisión y modo de revisiones múltiples · división de tráfico entre revisiones · secretos y variables de la aplicación · reglas de escalado HTTP y escalado a cero · escaladores de KEDA y cómo se declaran en Container Apps · registros y métricas del entorno para diagnosticar una revisión que no arranca

LABORATORIO: Publicar el servicio de búsqueda del caso en Container Apps con escalado HTTP y escalado a cero; dividir el tráfico 80/20 entre dos revisiones, provocar una revisión que no arranca y encontrar la causa en los registros del entorno

05 AKS: manifiestos, despliegue y diagnóstico

clúster de AKS con acceso al registro de contenedores · Deployment, Service y ConfigMap en manifiestos · Secret de Kubernetes y sus límites · sondas de disponibilidad y de vida · aplicar y revertir un despliegue con kubectl rollout · Container Insights y consultas sobre los registros del clúster · diagnóstico de ImagePullBackOff y CrashLoopBackOff · AKS, Container Apps o App Service: cuándo conviene cada uno

LABORATORIO: Desplegar la API del catálogo en AKS con manifiestos; provocar dos fallas (una imagen inexistente y una sonda mal configurada), diagnosticarlas con kubectl y Container Insights, documentar causa y corrección, y destruir el clúster al terminar

MÓDULO 3 DATOS PARA SOLUCIONES DE IA — dominio 2 del examen (25-30%) — 15 h · 5 sesiones

06 Cosmos DB for NoSQL desde el SDK: modelo, consultas, índices y consistencia

cuenta, base de datos y contenedor · elegir la clave de partición · operaciones con el SDK de Python: crear, leer, reemplazar y modificar parcialmente · consultas y su costo en unidades de solicitud (RU) · política de indexación: rutas incluidas, excluidas e índices compuestos · niveles de consistencia y su efecto en el costo y la latencia · errores 429 y reintentos

LABORATORIO: Cargar el historial de tickets de posventa en Cosmos DB, medir el costo en RU de cinco consultas del caso antes y después de ajustar la política de indexación, y justificar por escrito el nivel de consistencia elegido

07 Cosmos DB: embeddings, búsqueda vectorial y change feed processor

generar embeddings con un modelo desplegado en Azure · política de vectores e índice vectorial del contenedor · consultas por similitud con VectorDistance · combinar filtros y búsqueda vectorial en una misma consulta · change feed y el procesador de change feed · contenedor de concesiones y varias instancias del procesador · recalcular el embedding cuando cambia un documento

LABORATORIO: Agregar búsqueda vectorial al historial de tickets y un procesador de change feed que recalcula el embedding cuando un ticket cambia; demostrar que un ticket editado aparece en la búsqueda con su texto nuevo sin volver a cargar el contenedor

08 Azure Database for PostgreSQL desde código: esquema, índices y conexiones

servidor flexible y autenticación con Microsoft Entra ID · conexión desde Python con psycopg · esquema del catálogo de repuestos: tablas, claves e integridad · índices y su efecto en el plan de consulta · EXPLAIN ANALYZE para leer un plan · agrupación de conexiones y su efecto con muchos clientes · reintentos ante fallas transitorias

LABORATORIO: Cargar un catálogo de 50 000 piezas, mostrar con EXPLAIN ANALYZE la mejora de un índice sobre las tres consultas más usadas del caso y comparar latencia y errores con y sin agrupación de conexiones bajo 200 clientes concurrentes

09 PostgreSQL vectorial: pgvector, RAG y filtros de metadatos

habilitar la extensión pgvector · columna vector e inserción de embeddings · índices HNSW e IVFFlat y sus parámetros · recursos del servidor para cargas vectoriales: memoria, cómputo y almacenamiento · búsqueda vectorial con filtros de metadatos: marca, categoría y disponibilidad · recuperación para RAG: armar el contexto con las piezas encontradas · medir la recuperación contra un conjunto de consultas etiquetadas

LABORATORIO: Construir la búsqueda semántica del catálogo que encuentra la pieza aunque el cliente la describa con otras palabras, filtrada por marca y disponibilidad; medir aciertos en los cinco primeros resultados y latencia con y sin índice HNSW contra 30 consultas etiquetadas

10 Azure Managed Redis: caché, expiración, invalidación e índices vectoriales

crear una instancia de Azure Managed Redis y conectarse desde Python · patrón cache-aside · expiración de claves y políticas de expulsión · invalidación cuando cambia el dato de origen · índice vectorial en Redis y búsqueda por similitud · caché semántica de consultas parecidas · medir tasa de aciertos y latencia

LABORATORIO: Poner una caché cache-aside delante de la búsqueda del catálogo con invalidación disparada por un cambio de precio, y una caché semántica de consultas parecidas con índice vectorial; mostrar la tasa de aciertos y la latencia p95 antes y después, y destruir la instancia al terminar

MÓDULO 4 CONECTAR Y CONSUMIR SERVICIOS DE AZURE — dominio 3 del examen (20-25%) 12 h · 4 sesiones

11 Service Bus: colas y mensajes fallidos

espacio de nombres, colas y niveles del servicio · enviar y recibir con el SDK de Python · modos de recepción: bloqueo y lectura, y recepción con borrado · completar, abandonar y diferir un mensaje · cola de mensajes fallidos (dead-letter) y reproceso · detección de duplicados · un trabajador en Container Apps escalado por KEDA según la longitud de la cola

LABORATORIO: Llevar los pedidos del caso por una cola de Service Bus hasta un trabajador en Container Apps escalado por KEDA; enviar 500 pedidos y mostrar la curva de réplicas, y hacer que un pedido malformado termine en dead-letter con su motivo y se reprocese después de corregirlo

12 Service Bus: temas, suscripciones y reglas de filtro

temas y suscripciones · reglas de filtro SQL y de correlación · acciones de una regla sobre las propiedades del mensaje · dead-letter por suscripción y por expiración · propiedades del mensaje para enrutar · sesiones de mensajes para conservar el orden · cola o tema: cómo elegir

LABORATORIO: Publicar cada pedido una sola vez en un tema y enrutarlo con filtros a facturación, depósito y una suscripción de pedidos grandes; demostrar con contadores que cada pedido llegó exactamente a las suscripciones que le corresponden

13 Event Grid: eventos personalizados, filtros y reintentos

temas de sistema y temas personalizados · esquema de evento y CloudEvents · publicar eventos personalizados desde el SDK · suscripciones con filtros por tipo, por asunto y avanzados · política de reintentos y tiempo de vida del evento · destino para eventos no entregados · validación del punto de conexión de un webhook

LABORATORIO: Publicar los cambios de inventario como eventos personalizados con una suscripción que solo entrega los de stock bajo; con el destino caído, observar los reintentos y comprobar que los eventos no entregados quedan guardados para reprocesarlos

14 **Azure Functions: APIs serverless, triggers, bindings y despliegue**

planes de hospedaje y su efecto en el arranque · modelo de programación de Python para Functions · trigger HTTP para una API serverless · triggers de Service Bus, Event Grid y Cosmos DB · bindings de entrada y salida · configuración de la aplicación y conexiones con identidad · despliegue desde la línea de comandos y desde una tubería

LABORATORIO: Publicar una API serverless de consulta de piezas con trigger HTTP y una función que consume los eventos de stock bajo y escribe un aviso en Cosmos DB con un binding de salida; todas las conexiones por identidad, sin cadenas con secretos en la configuración

MÓDULO 5 SEGURIDAD, MONITOREO Y DIAGNÓSTICO — dominio 4 del examen (20-25%) 12 h · 4 sesiones

15 **Key Vault: secretos, acceso y rotación**

almacén de claves con control de acceso por roles · leer secretos desde el SDK con DefaultAzureCredential · versiones de un secreto · rotación de un secreto y su aviso por Event Grid · referencias a Key Vault desde App Service, Functions y Container Apps · caché del secreto en la aplicación y cuándo releerlo · registros de acceso al almacén

LABORATORIO: Rotar la clave de un proveedor externo del caso sin reiniciar el servicio: el aviso de vencimiento dispara la rotación y la aplicación toma la versión nueva; mostrar en el registro de acceso quién leyó el secreto y cuándo

16 **App Configuration: configuración centralizada**

almacén de App Configuration y acceso con identidad administrada · claves, etiquetas y configuración por entorno · referencias a Key Vault desde App Configuration · proveedor de configuración para Python · recarga dinámica con una clave centinela · banderas de funciones · qué va en App Configuration y qué va en Key Vault

LABORATORIO: Mover la configuración de los servicios del caso a App Configuration con etiquetas por entorno; activar una bandera que cambia el orden de los resultados de búsqueda y comprobar que el cambio se aplica sin volver a desplegar

17 **Trazas distribuidas con OpenTelemetry**

Application Insights basado en un área de trabajo de Log Analytics · Azure Monitor OpenTelemetry Distro para Python · spans automáticos y spans manuales · propagación del contexto por HTTP y por mensajería · atributos y eventos personalizados · muestreo y su efecto en lo que se ve · mapa de aplicación y búsqueda de transacciones

LABORATORIO: Instrumentar la API, el trabajador de pedidos y la función con OpenTelemetry, y seguir un pedido de punta a punta en una sola traza que cruza HTTP y Service Bus, con el tiempo de cada paso visible

18 **Consultas KQL para diagnosticar y alertar**

tablas de Log Analytics y de Application Insights · where, project, extend, summarize y join · percentiles de latencia y tasa de errores por operación · correlación de registros por identificador de operación · consultas sobre los registros de contenedores · alertas basadas en una consulta de registros · método de diagnóstico: síntoma, hipótesis, consulta y corrección

LABORATORIO: Resolver solo con consultas KQL tres fallas inyectadas en el caso (latencia en la búsqueda, pedidos que caen en dead-letter y errores 429 en Cosmos DB), documentar la consulta que encontró cada causa y dejar una alerta para cada una

MÓDULO 6 EXAMEN Y PROYECTO INTEGRADOR 6 h · 2 sesiones

19 **Preguntas de caso y simulacro cronometrado**

cómo se reparten los cuatro dominios en el examen · leer una pregunta de caso y encontrar el requisito que decide · descartar opciones por costo, seguridad y operación · simulacro cronometrado · diagnóstico por dominio · plan de repaso personal · logística del examen: agenda, identificación y supervisión

LABORATORIO: Rendir un simulacro cronometrado de 50 preguntas propias, revisar cada error con su dominio y escribir un plan de repaso para las dos semanas siguientes

20 **Defensa del proyecto integrador**

arquitectura final del caso · recorrido de una consulta y de un pedido de punta a punta · decisiones de cómputo, datos y mensajería justificadas · costo medido del entorno · identidades, secretos y configuración · evidencia de observabilidad y de diagnóstico · destrucción del entorno y cierre

`LABORATORIO:` Defender ante el grupo el servicio de posventa completo desplegado sin claves, mostrar la traza de un pedido y la consulta KQL que lo explica, presentar el costo medido y destruir todo el entorno al terminar

---

LO QUE VAS A USAR

Servicios y herramientas que aparecen en el temario del examen y en los laboratorios.

Contenedores

- Docker
- Azure Container Registry y ACR Tasks
- Azure App Service
- Azure Container Apps y KEDA
- Azure Kubernetes Service (AKS) y kubectl

Datos

- Azure Cosmos DB for NoSQL
- Azure Database for PostgreSQL con pgvector
- Azure Managed Redis
- Un modelo de embeddings desplegado en Azure

Integración

- Azure Service Bus
- Azure Event Grid
- Azure Functions

Seguridad y configuración

- Azure Key Vault
- Azure App Configuration
- Microsoft Entra ID e identidad administrada

Observabilidad

- OpenTelemetry
- Application Insights
- Log Analytics y KQL

Desarrollo

- Python y los SDK de Azure
- Bicep y la CLI de Azure
- Git y GitHub
- Azure Cost Management

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 laboratorio por sesión y un mismo caso que crece hasta el proyecto final.

01 **Clases en vivo por Zoom** 20 sesiones de 3 horas, 2 veces por semana. Se programa en vivo y se revisa el código de los laboratorios 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 **40% concepto / 60% taller** Se explica el servicio, sus límites y cómo lo pregunta el examen, y enseguida se construye la pieza del caso con su criterio de aceptación.

04 **Romper para diagnosticar** Varias sesiones provocan una falla a propósito —una imagen que no existe, un mensaje malformado, un secreto vencido— para practicar cómo se encuentra la causa con registros, trazas y consultas.

05 **Costo bajo control** Presupuesto y alertas desde la primera sesión, niveles chicos de cada servicio y recursos destruidos al cierre de cada práctica. El clúster de AKS, PostgreSQL y Redis viven solo durante la sesión que los usa.

06 **Simulacro y defensa** Una sesión de preguntas de caso con simulacro cronometrado y diagnóstico por dominio, y una sesión final para defender el proyecto ante el grupo.

Horarios · Primer grupo

Martes	19:00 - 22:00
Jueves	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 despliega infraestructura con código y conecta una aplicación Python a Azure en la primera hora.

Programación

- ✓ Python con soltura: funciones, clases, entornos virtuales y async
- ✓ Consumir APIs REST y trabajar con JSON
- ✓ SQL básico: consultas, uniones e índices
- ✓ Git y GitHub a nivel diario

Contenedores y Azure

- ✓ Construir y ejecutar una imagen de Docker
- ✓ Crear recursos y grupos de recursos en Azure
- ✓ Asignar roles a usuarios e identidades
- ✓ Usar la CLI de Azure

Certificación

- ✓ Perfil de certificación de Microsoft creado, preferentemente con cuenta personal
- ✓ Documento de identidad válido
- ✓ Microsoft no declara requisitos previos para rendir el examen

Tiempo y cuenta

- ✓ Seis horas de clase por semana
- ✓ Entre cuatro y seis horas de laboratorio
- ✓ Suscripción propia de Azure con tarjeta asociada
- ✓ 10 semanas de compromiso

Costos de herramientas que asumes tú `US$40 a 80 aprox.`

Estos costos NO están incluidos en el precio del programa. El examen se agenda, se paga y se rinde directamente con el proveedor de evaluación de Microsoft; nosotros preparamos para él pero no lo administramos.

- — Aparte de este monto: la tasa del examen AI-200, que fija Microsoft según el país donde lo rindas. Consulta el precio en la página oficial del examen antes de inscribirte.
- — Suscripción propia de Azure para los laboratorios: registro de contenedores, App Service, Container Apps, un clúster de AKS por una sesión, Cosmos DB, PostgreSQL, Azure Managed Redis, Service Bus, Event Grid, Functions y un modelo de embeddings, consumidos por uso
- — Los servicios más caros (el clúster de AKS, PostgreSQL y Redis) se crean al empezar la sesión que los usa y se destruyen al terminarla
- — Se configura presupuesto y alertas al 50, 75 y 90 % en la sesión 01, y cada laboratorio termina destruyendo lo que creó

PRECIO

60 horas para construir en Azure lo que el examen evalúa.

US$150

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

- ✓ 20 sesiones en vivo por Zoom (60 horas lectivas)
- ✓ Grabación de todas las clases con acceso por 12 meses
- ✓ Repositorio del caso con el punto de partida de cada laboratorio y su criterio de aceptación
- ✓ Datos de prueba del caso: catálogo de piezas, tickets, pedidos y consultas etiquetadas
- ✓ Plantillas de Bicep del entorno de cada bloque
- ✓ Simulacro cronometrado de 50 preguntas propias con diagnóstico por dominio
- ✓ Los 3 certificados progresivos de preparación
- ✓ Canal privado del grupo para dudas entre sesiones
- ✓ Actualizaciones del material por 12 meses

No incluye

- — La tasa del examen AI-200: se agenda, se paga y se rinde directamente con el proveedor de evaluación de Microsoft
- — El consumo de tu suscripción de Azure para los laboratorios (aprox. US$40 a 80 en todo el programa)

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: 20 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 Martes y jueves 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.

Venía preparando el AZ-204. ¿Qué hago ahora?

El AZ-204 se retiró el 31 de julio de 2026, según la lista de exámenes retirados que publica Microsoft. Para quien venía preparándolo, el examen de desarrollador de Azure vigente es el AI-200. No son el mismo examen: el AI-200 tiene su propio temario de cuatro dominios, y este programa sigue ese temario, no el del AZ-204. Antes de inscribirte, revisa la guía de estudio del AI-200 en la página oficial de Microsoft.

¿En qué se diferencia de la preparación para el AI-103?

Son dos exámenes distintos con poco en común. El AI-103 evalúa cómo se construyen aplicaciones y agentes con modelos en Microsoft Foundry. El AI-200 evalúa la nube alrededor de esas aplicaciones: contenedores, bases de datos con búsqueda vectorial, mensajería, secretos, configuración y trazas. Casi no hay sesiones que se repitan entre los dos programas; el punto de contacto es la búsqueda vectorial, que aquí se trata desde las bases de datos y no desde los servicios de IA. Cada programa es completo por sí mismo: no hay uno previo ni descuentos entre ellos.

¿Qué nivel necesito?

Experiencia desarrollando aplicaciones, soltura con Python, nociones de SQL y familiaridad con Azure y con Docker. Si nunca creaste un recurso en Azure o nunca construiste una imagen de contenedor, este programa te va a quedar grande. Los requisitos están detallados más arriba.

¿En qué lenguaje son los laboratorios?

En Python, con los SDK de Azure para Python. El examen evalúa cómo se programa contra cada servicio: antes de inscribirte, revisa en la guía de estudio oficial del AI-200 cómo presenta los fragmentos de código.

¿El examen está incluido en el precio?

No. La tasa del examen la cobra Microsoft a través de su proveedor de evaluación y varía según el país donde lo rindas. Tú agendas, pagas y rindes el examen directamente con ellos. Nosotros te preparamos para ese examen, pero no lo administramos ni lo revendemos.

¿Este curso es oficial de Microsoft?

No. Reactor Courses no es partner de formación de Microsoft, este programa no es material oficial y no tenemos ningún aval ni acreditación de Microsoft. Es un programa de preparación construido por nosotros a partir del temario público del examen, con un caso y laboratorios propios.

¿Cuánto voy a gastar en Azure?

Estimamos unos US$40 a 80 de consumo en todo el programa, con precios de lista. Se configura un presupuesto con alertas en la sesión 01, se usan los niveles más chicos de cada servicio, el clúster de AKS, PostgreSQL y Redis se crean y se destruyen dentro de la sesión que los usa, y cada práctica termina eliminando lo que creó. La tasa del examen va aparte.

¿Cómo es el examen? ¿Se puede rendir en español?

Según la página de Microsoft consultada al preparar este programa, el AI-200 dura 120 minutos, es supervisado, se aprueba con 700 puntos o más y se ofrece en español, entre otros idiomas. Microsoft actualiza primero la versión en inglés y las traducidas después: si el temario cambia cerca de tu fecha, conviene revisar en qué versión lo rindes.

¿Qué pasa si falto a una clase?

La grabación queda disponible el mismo día y el laboratorio de cada sesión tiene su punto de partida en el repositorio, así que puedes ponerte al día antes de la siguiente. Los laboratorios se acumulan sobre el mismo caso: faltar a varias seguidas sin recuperarlas te deja sin las piezas que usa el proyecto final.

¿Me aseguran que apruebo?

No, y desconfía de quien lo prometa. Lo que te damos es cobertura completa del temario publicado, un laboratorio por sesión, preguntas de caso y un simulacro con diagnóstico por dominio. Aprobar depende también de tu experiencia previa y del día.

¿Qué pasa si el examen cambia mientras curso el programa?

Microsoft actualiza el temario de sus exámenes periódicamente y en 2026 ya retiró varios. Verificamos el temario publicado antes de que empiece cada grupo y tienes 12 meses de actualizaciones del material. Si el examen cambiara mientras cursas el programa, te avisamos y ajustamos el contenido afectado.

¿Qué métodos de pago aceptan?

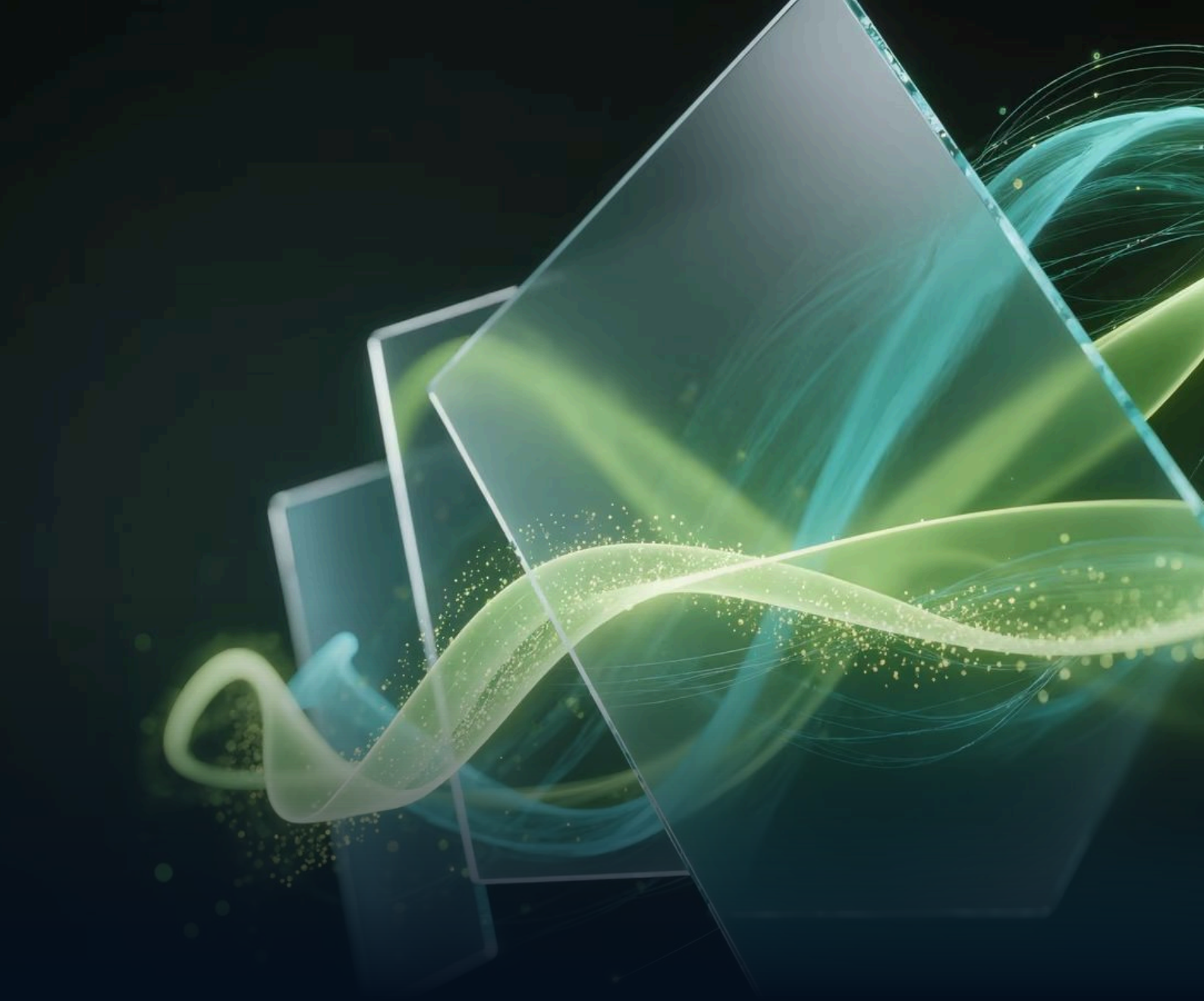
El pago lo procesa Paddle y acepta Tarjeta de crédito, Tarjeta de débito internacional, PayPal, Apple Pay, Google Pay. El cobro es en dólares. Importante: no procesamos Yape, Plin, OXXO, Mercado Pago ni transferencias locales. Si tu banco rechaza compras internacionales, habilítalas antes de pagar o 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/ai-200-azure-ai-cloud-developer.

`reactorcourses.com/programas/ai-200-azure-ai-cloud-developer`

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.