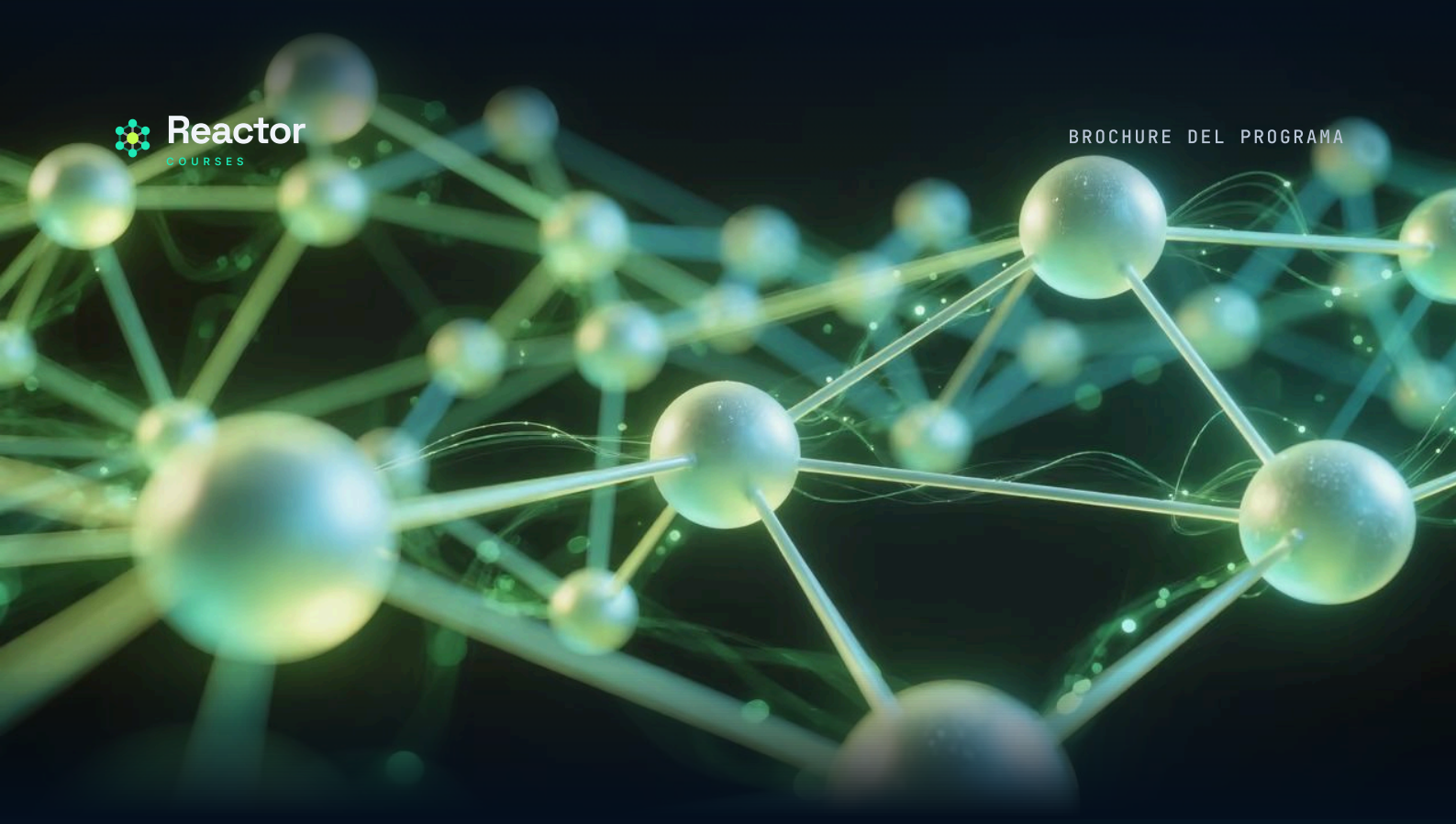


Reactor
COURSES

BROCHURE DEL PROGRAMA

ARQUITECTURA E INGENIERÍA · INTELIGENCIA ARTIFICIAL

El programa más completo de la escuela

Enterprise IA Agentic Developer

Todo el ciclo de los agentes de IA, de tu editor a producción en Azure y AWS

Diseña, construye, evalúa, asegura y despliega aplicaciones nativas de IA (AI-Native) con agentes de IA sobre Azure y AWS.

Lo que vas a lograr Aprendes a construir aplicaciones en las que la IA no solo responde: consulta datos, usa herramientas y completa tareas, y a ponerlas en producción en Azure y AWS.

Claude Code · OpenAI Agents SDK · RAG · MCP · A2A · Microsoft Foundry · Bedrock AgentCore · Terraform · CI/CD

horas en vivo	sesiones	semanas	nubes (Azure y AWS)
162	54	18	2

INICIO
6 de febrero de 2027

PRECIO
US$350 pago único · USD

MODALIDAD
Clases en vivo por Zoom · Avanzado

reactorcourses.com/programas/agentic-ai-application-engineering

EL PROGRAMA

Dos competencias en un mismo programa: construir con agentes y construir agentes.

Diseña, construye, evalúa, asegura y despliega aplicaciones nativas de IA (AI-Native) con agentes de IA sobre Azure y AWS. Un programa de **162 horas** para ingenieros que ya escriben código y quieren construir sistemas con agentes, no demos de chatbot.

Duración	162 horas · 54 sesiones	Calendario	18 semanas · 3 sesiones por semana de 3 h
Modalidad	Clases en vivo por Zoom	Grabaciones	Disponibles por 12 meses
Credenciales	4 certificados con insignia de Reactor Courses	Nivel	Avanzado
Inicio	6 de febrero de 2027	Horario	Martes y jueves, 19:00–22:00 · Sábado, 07:00–10:00 (GMT-5)
Precio	US$350 · pago único		

Una es usar agentes de IA con disciplina para construir software: requisitos, código, pruebas, revisión y despliegue. La otra es diseñar y operar aplicaciones en las que el agente forma parte del sistema en producción.

01

Construir CON agentes

Ingeniería de software asistida por IA

Usar Claude Code de forma disciplinada en el ciclo completo de ingeniería: requisitos, especificaciones, arquitectura, código, pruebas, revisión, infraestructura y CI/CD. No es autocompletado: es un flujo de trabajo con puntos de control humanos y evidencia.

- ✓ Ingeniería de contexto (Context Engineering) e instrucciones persistentes del repositorio
- ✓ Desarrollo guiado por especificaciones (Spec-Driven Development) y trazabilidad requisito → código → prueba
- ✓ Gobernanza, calidad y seguridad del código generado por IA
- ✓ Arquitectura C4, ADR y contratos OpenAPI

02

Construir agentes

Ingeniería de aplicaciones con agentes

Diseñar aplicaciones donde el agente forma parte del sistema de negocio en ejecución: razona, elige herramientas, consulta conocimiento, mantiene contexto, ejecuta acciones y colabora con otros agentes — dentro de límites explícitos.

- ✓ Herramientas (tools), llamadas a funciones y salida estructurada con contratos
- ✓ RAG, búsqueda híbrida, memoria de corto y largo plazo
- ✓ MCP, A2A y sistemas de varios agentes con control de costos
- ✓ Barreras de seguridad (guardrails), modelado de amenazas, aprobación humana (HITL), evaluaciones y trazas

> Un agente no es un prompt conectado a un modelo. Es un componente de software que opera dentro de límites técnicos, funcionales, de seguridad y de negocio explícitos.

PARA QUIÉN ES

¿Es el programa indicado para ti?

Está pensado para desarrolladores con experiencia que quieren llevar agentes de IA a producción. Si ya programas con soltura, aquí aprendes a construirlos, medirlos y operarlos en Azure y AWS.

Es para ti si

- ✓ Desarrolladores y ingenieros de software o sistemas con experiencia real escribiendo código.
- ✓ Arquitectos y líderes técnicos que deben decidir dónde entra un agente y dónde no.
- ✓ Ingenieros de nube, DevOps y plataforma que van a operar estos sistemas en producción.
- ✓ Ingenieros de datos y de automatización de pruebas con base sólida de programación.
- ✓ Estudiantes de últimos ciclos de carreras tecnológicas que ya programan con soltura.

Te conviene otro programa

- – Personas sin experiencia previa programando. Este programa no enseña a programar.
- – Quien busque un curso de "prompt engineering" o de uso de ChatGPT.
- – Quien no pueda dedicar entre 12 y 15 horas por semana durante 18 semanas.
- – Quien quiera resultados sin escribir código ni desplegar infraestructura.
- – Quien no pueda asumir el costo propio de herramientas y consumo de nube del laboratorio.

EL RESULTADO

Terminas con un sistema desplegado en dos nubes, no con un certificado de asistencia.

El proyecto final es un sistema desplegado, documentado y auditable. Puedes usarlo como material para una entrevista técnica o para sustentar una propuesta interna en tu empresa.

01 **Repositorio profesional** Código Python, FastAPI, Pydantic, pruebas unitarias y de integración.

02 **Arquitectura documentada** Diagramas C4 de contexto y de contenedores, ADR de cada decisión, contratos OpenAPI.

03 **Agentes en producción** 3 flujos de trabajo, al menos 4 herramientas (una que modifica datos), sesiones y memoria.

04 **RAG en producción** Ingesta, fragmentación (chunking), embeddings, búsqueda híbrida y evaluación de la recuperación.

05 **Capa para varios modelos** El mismo caso corriendo sobre GPT, Gemini, Claude y un modelo abierto.

06 **Seguridad demostrable** Modelo de amenazas, barreras de seguridad (guardrails), autorización determinística, aprobación humana (HITL) y registro de auditoría.

07 **Conjunto de evaluación de 50 casos** 20 del camino feliz, 10 casos límite, 5 no autorizados, 5 de inyección de prompts, 5 de mal uso de herramientas y 5 de recuperación.

08 **Observabilidad y FinOps** Trazas correlacionadas, métricas y costo por tarea proyectado a 100 000 usuarios.

09 **Despliegue en Azure** Microsoft Foundry, Entra ID, Key Vault, AI Search y Application Insights.

10 **Despliegue en AWS** Bedrock AgentCore Runtime, Gateway e Identity, IAM, Secrets Manager, CloudWatch.

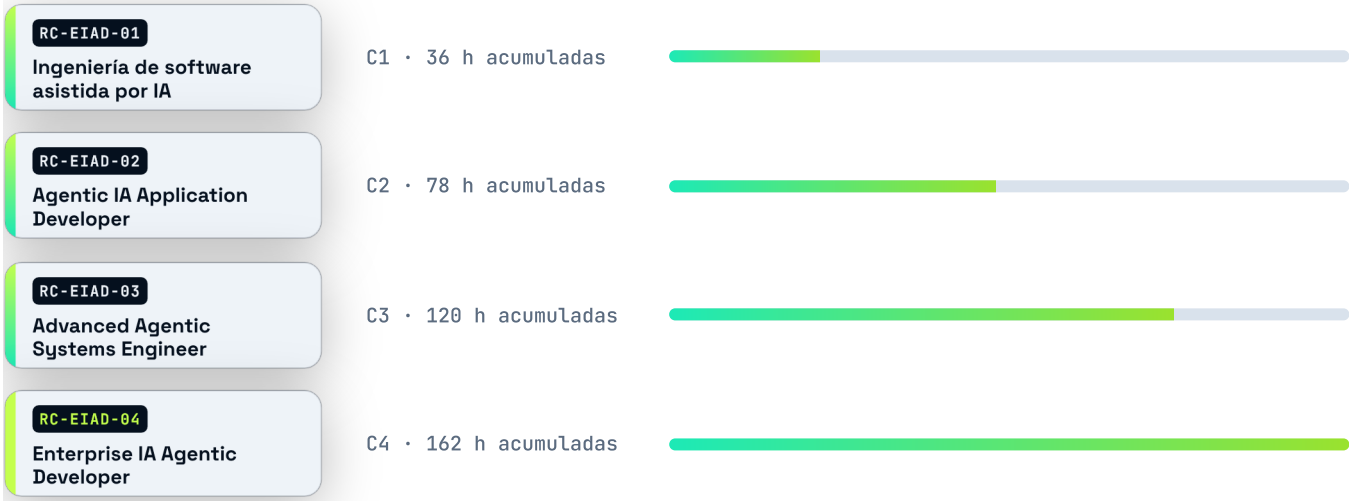
11 **Infraestructura como código** Docker, Terraform y pipelines de CI/CD en los que las evaluaciones deciden si se despliega.

12 **Defensa técnica** Día de demostración: sustentas tu arquitectura y respondes preguntas de un comité.

RUTA DE CERTIFICACIÓN

Cuatro credenciales progresivas, no una al final.

Cada etapa cierra un bloque completo de competencias. Si algo te obliga a pausar en la semana 10, ya tienes dos credenciales defendibles en tu CV.



CERTIFICACIÓN INTEGRAL

Enterprise IA Agentic Developer (162 horas)

Requiere aprobar el proyecto final y la defensa técnica. 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 54 sesiones, abiertas y sin letra chica.

162 horas repartidas en 14 módulos. Así sabes qué verás, qué practicarás y qué entregarás en cada una de las 18 semanas.

MÓDULO 1 INGENIERÍA DE SOFTWARE NATIVA DE IA Y CON AGENTES — 6 h · 2 sesiones

01 Software tradicional, IA generativa y aplicaciones nativas de IA

foundation models · inference vs training · token · context window · sampling · temperature · top-p · non-determinism · hallucination · model latency

LABORATORIO: Clasificar 5 casos de negocio y justificar la arquitectura

02 Anatomía del entorno de ejecución de un agente

model · instructions · user input · runtime context · state · session · memory · tool registry · guardrails · tracing

LABORATORIO: lienzo de límites del agente del proyecto final

MÓDULO 2 CLAUDE CODE Y DESARROLLO DE SOFTWARE ASISTIDO POR IA — 12 h · 4 sesiones

03 Agentes de programación y Claude Code

coding agent vs autocomplete · repository awareness · filesystem · shell · code edits · tests · plan/execute/review loops · permission model · puntos de control humanos · repository map

LABORATORIO: incorporar un repositorio existente con Claude Code

04 Ingeniería de contexto

prompt vs context · static vs dynamic context · authoritative sources · relevance · freshness · conflicts · repository rules · coding standards · architectural rules · testing rules

LABORATORIO: crear archivo de instrucciones del proyecto

05 Flujo de desarrollo asistido por agentes

task decomposition · small batches · reproducible bug · failing test first · patch · regression · refactoring · documentation · PR preparation

LABORATORIO: resolver un error con evidencia completa

06 Gobernanza, calidad y seguridad del código generado por IA

dependency hallucination · insecure generation · secret exposure · destructive commands · false test confidence · stale libraries · code review · lint · type check · unit tests

LABORATORIO: control de calidad local

MÓDULO 3 DESARROLLO GUIADO POR ESPECIFICACIONES Y ARQUITECTURA — 12 h · 4 sesiones

07 Ingeniería de requisitos

problem statement · stakeholder · goal · non-goal · scope · assumption · constraint · functional requirement · NFR · persona

LABORATORIO: backlog verificable del proyecto final

08 PRD y especificación técnica

problem · users · value · success metrics · scope · workflow · constraints · risks · components · interfaces

09 Arquitectura de software para aplicaciones con agentes

C4 Context · C4 Container · C4 Component · system boundaries · API boundary · agent service · tool layer · domain layer · model adapter · retrieval layer

LABORATORIO: C4 del proyecto final

10 **ADR, OpenAPI y contratos**

status · context · decision · alternatives · consequences · risks · rollback · framework · multi-model strategy · RAG

`LABORATORIO:` contrato de API que luego se convertirá en herramienta

MÓDULO 4 INGENIERÍA DE AGENTES — 15 h · 5 sesiones

11 **Primer agente**

Agent · instructions · model · Runner · run lifecycle · input · result · context · structured output · errors

`LABORATORIO:` agente sin herramientas y conjunto mínimo de pruebas

12 **Salidas estructuradas**

free text vs structured output · JSON Schema · Pydantic · enums · required · optional · nested objects · validation · semantic validation · schema evolution

`LABORATORIO:` intención y parámetros tipados

13 **Llamadas a funciones y diseño de herramientas**

function declaration · name · description · parameters · required fields · tool choice · execution · result · continuation · single responsibility

`LABORATORIO:` 3 herramientas de solo lectura

14 **Herramientas transaccionales**

query vs command · side effects · idempotency · retry · timeout · duplicate calls · compensating action · audit · transaction boundaries · confirmation

`LABORATORIO:` herramienta que modifica datos, con clave de idempotencia

15 **Sesiones, estado y contexto de ejecución**

stateless · stateful · conversation history · session state · user state · application state · tool state · runtime context · correlation id · expiration

`LABORATORIO:` sesiones para 2 usuarios y prueba de aislamiento

MÓDULO 5 INGENIERÍA CON VARIOS MODELOS — 9 h · 3 sesiones

16 **Abstracción del proveedor de modelos**

provider · model id · endpoint · credential · capability · retry · quota · rate limit · adapter · dependency inversion

`LABORATORIO:` interfaz común y adaptadores

17 **GPT, Gemini, Claude y un modelo abierto, frente a frente**

instructions · structured output · tools · streaming · multimodal · reasoning · context · latency · cost · rate limits

`LABORATORIO:` mismo dataset y misma tarea con 4 modelos

18 **Enrutamiento, respaldo y selección de modelo**

task · complexity · risk · modality · latency SLA · budget · context length · availability · timeout · 429

`LABORATORIO:` enrutador de modelos

MÓDULO 6 RAG, CONOCIMIENTO Y MEMORIA

12 h · 4 sesiones

19 **Ingesta, fragmentación y embeddings**

document source · parser · normalization · metadata · chunk · overlap · token length · section-aware chunking · semantic chunking · embedding

LABORATORIO: pipeline de ingesta

20 **Recuperación avanzada**

lexical · BM25 concept · semantic · hybrid · metadata filters · query rewriting · multi-query · reranking · threshold · top-k

LABORATORIO: comparar 3 estrategias de recuperación

21 **Generación fundamentada y evaluación de RAG**

grounding · answerability · evidence · citations · abstention · conflicting sources · stale information · hallucination

LABORATORIO: endpoint RAG con evidencia

22 **Memoria del agente**

working · short-term · long-term · semantic · episodic · preferences · write policy · read policy · retention · deletion

LABORATORIO: CRUD de memoria con política explícita

MÓDULO 7 MCP, A2A E INTEGRACIÓN EMPRESARIAL

12 h · 4 sesiones

23 **Fundamentos de MCP**

client · server · tools · resources · prompts · discovery · transport · remote/local · lifecycle concept · REST vs MCP

LABORATORIO: consumir un servidor MCP

24 **Construcción y seguridad de un servidor MCP**

server boundary · tool registry · schema · description · output · errors · versioning · authentication · authorization · allowlist

LABORATORIO: servidor MCP del proyecto final

25 **APIs empresariales e identidad**

REST · OpenAPI · API key · OAuth 2.0 · OIDC · client credentials · delegated identity concept · sync · async · queue

LABORATORIO: herramienta protegida

26 **A2A**

agent interoperability · remote agent · discovery · Agent Card concept · capabilities · delegation · tasks · trust · authentication · timeout

LABORATORIO: contrato de integración entre agentes remotos

MÓDULO 8 SISTEMAS DE VARIOS AGENTES

12 h · 4 sesiones

27 **Decidir entre un agente y varios**

context isolation · permission separation · specialization · ownership · provider differences · independent lifecycle · latency · tokens · complexity · debugging

LABORATORIO: revisión de arquitectura

28 **Enrutador, supervisor y agentes como herramientas**

deterministic · model-based · confidence · fallback · plan · delegate · aggregate · bounded specialist · explicit invocation · controlled context

LABORATORIO: supervisor y 2 especialistas

29 **Traspaso, planificador y ejecutor, y agentes en paralelo**

transfer of control · context filtering · ownership · planner · executor · validation · replan · fan-out · fan-in · timeout

`LABORATORIO:` implementar 2 patrones

30 **Confiabilidad de los sistemas de varios agentes**

ping-pong · infinite delegation · circular dependencies · repeated tool call · state corruption · runaway cost · max turns · max handoffs · max tokens · time budget

MÓDULO 9 SEGURIDAD, BARRERAS Y APROBACIÓN HUMANA — 9 h · 3 sesiones

31 **Modelado de amenazas para agentes**

credentials · PII · memory · prompt · tool · data · logs · prompt injection · indirect prompt injection · jailbreak

`LABORATORIO:` modelo de amenazas

32 **Identidad, autorización y barreras de seguridad**

user · app · agent · tool/service · RBAC · ABAC concept · ownership · delegated access · least privilege · input

`LABORATORIO:` bloquear una llamada a herramienta no autorizada

33 **Aprobación humana (human-in-the-loop)**

read-only · low-risk write · destructive · financial · irreversible · high-impact · action · params · requester · risk

`LABORATORIO:` pausar, aprobar y reanudar

MÓDULO 10 EVALUACIONES, OBSERVABILIDAD Y OPERACIÓN DE AGENTES — 9 h · 3 sesiones

34 **Ingeniería de evaluaciones**

id · input · context · expected behavior · required tools · prohibited tools · expected facts · risk · tags · exact

`LABORATORIO:` dataset inicial de 20 casos

35 **Métricas y regresión**

Task Success Rate · Tool Selection Accuracy · Tool Argument Accuracy · Tool Execution Success · Groundedness · Hallucination Rate · Safety Violation Rate · Escalation Rate · Latency · Tokens

`LABORATORIO:` pipeline de regresión

36 **Trazas, registros y operación de agentes**

trace id · span id · parent · model call · tool · retrieval · guardrail · approval · structured logging · correlation

MÓDULO 11 DOCKER, TERRAFORM Y CI/CD — 9 h · 3 sesiones

37 **Docker**

image · layer · container · Dockerfile · base · dependencies · COPY · ENTRYPOINT/CMD · env · port

`LABORATORIO:` contenedor del servicio del agente

38 **Terraform**

provider · resource · data · variable · output · local · module · state · plan · apply

`LABORATORIO:` infraestructura reproducible

39 **CI/CD para aplicaciones con agentes**

task success threshold · safety threshold · latency max · cost regression max · tool accuracy

`LABORATORIO:` GitHub Actions

MÓDULO 12 MICROSOFT AZURE Y MICROSOFT FOUNDRY

15 h · 5 sesiones

40 **Bases de Azure e identidad**

tenant · subscription · resource group · resource · Entra ID · user · group · service principal concept · managed identity · RBAC

LABORATORIO: entorno aislado

41 **Fundamentos de Microsoft Foundry**

Foundry platform · projects · model catalog · model endpoint · identity · prompt-based agent · code-based agent · tools · eval · observability

LABORATORIO: conectar la aplicación a un modelo de Foundry

42 **Agentes alojados en Foundry**

bring-your-own-code · framework-neutral hosting · source/container packaging · managed endpoint · scaling · dedicated Entra identity · session persistence · observability · configuration · dependencies

LABORATORIO: empaquetar el servicio del agente

43 **Azure AI Search, seguridad y observabilidad**

index · fields · vectors concept · metadata · filters · Managed Identity · Key Vault · RBAC · Azure Monitor · Application Insights

LABORATORIO: telemetría de punta a punta

44 **Despliegue en Azure**

functional · auth · unauthorized · tool · trace · rollback · cost

MÓDULO 13 AWS BEDROCK Y AGENTCORE

15 h · 5 sesiones

45 **Bases de AWS e IAM**

account · region · resource · organization concept · IAM principal · user · role · policy · action · resource

LABORATORIO: permisos mínimos del sandbox

46 **Amazon Bedrock**

model platform · model providers · inference · model access · endpoint/invocation concept · model selection · Bedrock vs AgentCore

LABORATORIO: invocación de modelos y comparación

47 **AgentCore Runtime**

agent runtime · framework agnostic · model flexibility · packaging · invocation · session isolation · MCP · A2A · long-running workload concept · runtime boundary

LABORATORIO: agente listo para desplegar

48 **AgentCore Gateway e Identity**

agentic traffic · APIs · tools · MCP · ingress auth · egress auth · model routing concept · A2A passthrough concept · workload identity · outbound auth

LABORATORIO: agente → Gateway → herramienta

49 **Despliegue y observabilidad en AWS**

image · AgentCore Runtime · IAM · configuration · secrets · invocation · CloudWatch · logs · metrics · tracing integration

MÓDULO 14 **PROYECTO FINAL EN DOS NUBES** 15 h · 5 sesiones

50 **Comité de revisión de arquitectura**
business problem · PRD · C4 · ADR · agents · tools · data · model strategy · RAG · security

51 **Refuerzo de confiabilidad, seguridad y costo**
timeout · retry · circuit breaker · fallback · idempotency · max turns · threat closure · authz · HITL · audit

52 **Validación en las dos nubes**
endpoint · identity · tools · logs · trace · endpoint · identity · tools · logs · trace

53 **Preparación del portafolio**
problem · architecture · capabilities · setup · security · evals · cost · deployment · limitations · roadmap

54 **Día de demostración y defensa técnica**
¿qué pasa si la tool se llama dos veces? · ¿dónde se valida autorización? · ¿cómo evita memory leakage? · ¿qué ocurre ante rate limit? · ¿cuánto cuesta la operación? · ¿cómo detecta regresión? · ¿qué cambia Azure vs AWS? · ¿por qué ese modelo? · ¿por qué multi-agent? · ¿qué simplificaría?

HERRAMIENTAS

Todo lo que vas a tocar con las manos.

Ingeniería
Python · FastAPI · Pydantic · pytest · HTTPX · Git · GitHub · OpenAPI

Desarrollo asistido por IA
Claude Code · Context Engineering · Spec-Driven Development · Subagentes

Ingeniería de agentes
OpenAI Agents SDK · Function calling · Structured outputs · Sessions

Modelos
OpenAI GPT · Google Gemini · Anthropic Claude · Modelo abierto

Interoperabilidad
HTTP/REST · OpenAPI · MCP · A2A

Conocimiento
Embeddings · Vector search · Hybrid search · Reranking · RAG

DevOps
Docker · Terraform · GitHub Actions

Observabilidad
Logs estructurados · Tracing · Métricas · OpenTelemetry

Microsoft Azure
Microsoft Foundry · Foundry Agent Service · Entra ID · Managed Identity · Key Vault · AI Search · Application Insights · ACR

Amazon Web Services
Bedrock · AgentCore Runtime · AgentCore Gateway · AgentCore Identity · IAM · Secrets Manager · ECR · CloudWatch

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, laboratorio permanente y evidencia en cada entrega.

01 **Clases en vivo por Zoom** 54 sesiones de 3 horas, 3 veces por semana. Cámara y micro abiertos para preguntar: no es un webinar, es una clase.

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** La mayor parte del tiempo escribes código, rompes cosas y las arreglas. Cada sesión cierra con un entregable concreto.

04 **Controles de calidad, no notas blandas** Cinco compuertas: especificación lista, código listo, agente listo, seguridad lista y listo para producción. No avanzas sin evidencia.

05 **Proyecto final en equipo, defensa individual** Equipos de 5 con roles rotativos: ingeniería de agentes, backend, RAG, nube y DevOps, y seguridad y evaluaciones. Cada uno defiende lo suyo.

06 **Evidencia sobre opinión** Cada decisión se justifica con pruebas, evaluaciones, trazas, mediciones comparativas, métricas y ADR. Nada se aprueba porque "funciona en mi máquina".

Horarios · Primer grupo

Martes	19:00 - 22:00
Jueves	19:00 - 22:00
Sábado	07:00 - 10:00

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

Las clases que caigan en feriado se reprograman. El calendario definitivo se entrega al confirmar la matrícula.

REQUISITOS

Lo que necesitas saber antes de la sesión 01.

No hay nivelación previa. La sesión 01 asume todo esto resuelto.

Programación

- ✓ Variables, funciones, clases
- ✓ Excepciones, módulos y paquetes
- ✓ Estructuras de datos
- ✓ Asincronía a nivel introductorio

Python

- ✓ Ejecutar scripts
- ✓ Entornos virtuales (venv)
- ✓ Instalar dependencias
- ✓ Leer y modificar código
- ✓ Trabajar con JSON
- ✓ Consumir APIs HTTP

Web y APIs

- ✓ HTTP y verbos REST
- ✓ Cabeceras y códigos de estado
- ✓ JSON
- ✓ Autenticación básica

Git

- ✓ clone, branch, commit
- ✓ pull, push, merge
- ✓ pull request

Nube

- ✓ Cómputo y almacenamiento
- ✓ Redes básicas
- ✓ Identidad
- ✓ Secretos y registros (logs)

Costos de herramientas que asumes tú `US$20 a 50 aprox.`

Estos costos NO están incluidos en el precio del programa. Se declaran por transparencia: cada alumno administra sus propias cuentas y presupuestos. La suscripción a Claude sí está incluida: la da la escuela durante todo el programa.

- — Consumo de APIs de OpenAI, Gemini y Claude en laboratorios (presupuesto controlado)
- — Azure y AWS: se trabaja sobre los niveles gratuitos (free tier) y créditos educativos cuando aplican

PRECIO

162 horas, dos nubes y un sistema desplegado.

US$350

Pago único · USD · Incluye impuestos

Primer grupo · inicia el 6 de febrero de 2027

Incluye

- ✓ 54 sesiones en vivo por Zoom (162 horas lectivas)
- ✓ Grabación de todas las clases con acceso por 12 meses
- ✓ Laboratorios, datasets y repositorios de referencia
- ✓ Revisión de entregables en los 5 controles de calidad
- ✓ Proyecto final en dos nubes con acompañamiento técnico
- ✓ Los 4 certificados progresivos y la certificación integral
- ✓ Canal privado del grupo para dudas entre sesiones
- ✓ Actualizaciones del material por 12 meses
- ✓ Suscripción a Claude (plan Pro o equivalente, incluye Claude Code) durante todo el programa: te la damos al inscribirte

No incluye

- – Consumo de APIs y de nube propio en los laboratorios (aprox. US$20 a 50)

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: 54 sesiones de 3 horas, 3 veces por semana. Todas quedan grabadas y disponibles por 12 meses, así que puedes recuperar una clase perdida o repasar un laboratorio.

¿Qué pasa si el horario no me calza por mi zona horaria?

El horario está referido a GMT-5 (Perú, Colombia, Ecuador y Panamá; en Ciudad de México es una hora menos). En Panamá es la misma hora; en Ciudad de México y Centroamérica las sesiones empiezan una hora antes; en Bolivia, Venezuela y el Caribe, una hora después; en Argentina, Uruguay, Paraguay y Brasil, dos horas después, y en Chile, una o dos según la época del año. Puedes seguir el programa por las grabaciones y usar el canal del grupo para tus dudas, pero pierdes la interacción en vivo. Escríbenos antes de comprar si tu diferencia horaria es mayor a tres horas.

¿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.

¿Puedo pagar en cuotas?

Por ahora el programa se cobra en un pago único de US$350. Algunos bancos permiten diferir la compra en cuotas desde su propia app o banca en línea; eso depende de tu emisor, no de nosotros.

¿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. Si necesitas datos fiscales de empresa, cárgalos en el campo correspondiente de la ventana de 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 y nadie llegue a ese punto.

¿Necesito saber programar?

Sí, y con soltura. El programa asume Python, Git, HTTP/REST y conceptos básicos de nube desde la sesión 01. No hay nivelación previa. Revisa la sección de requisitos con honestidad antes de inscribirte.

¿Cuánto tiempo debo dedicarle por semana?

Nueve horas de clase más entre tres y seis horas de laboratorio y entregables. Considera entre 12 y 15 horas semanales durante 18 semanas.

¿Qué costos adicionales tengo que asumir?

La suscripción a Claude (plan Pro o equivalente, con Claude Code) te la damos nosotros durante todo el programa. Aparte quedan unos US$20 a 50: el consumo de APIs en los laboratorios y los recursos de Azure y AWS. Trabajamos sobre los niveles gratuitos (free tier) y créditos educativos cuando aplican, y el programa incluye disciplina de presupuestos y limpieza de recursos para que no se te dispare la cuenta.

¿Los certificados tienen validez universitaria?

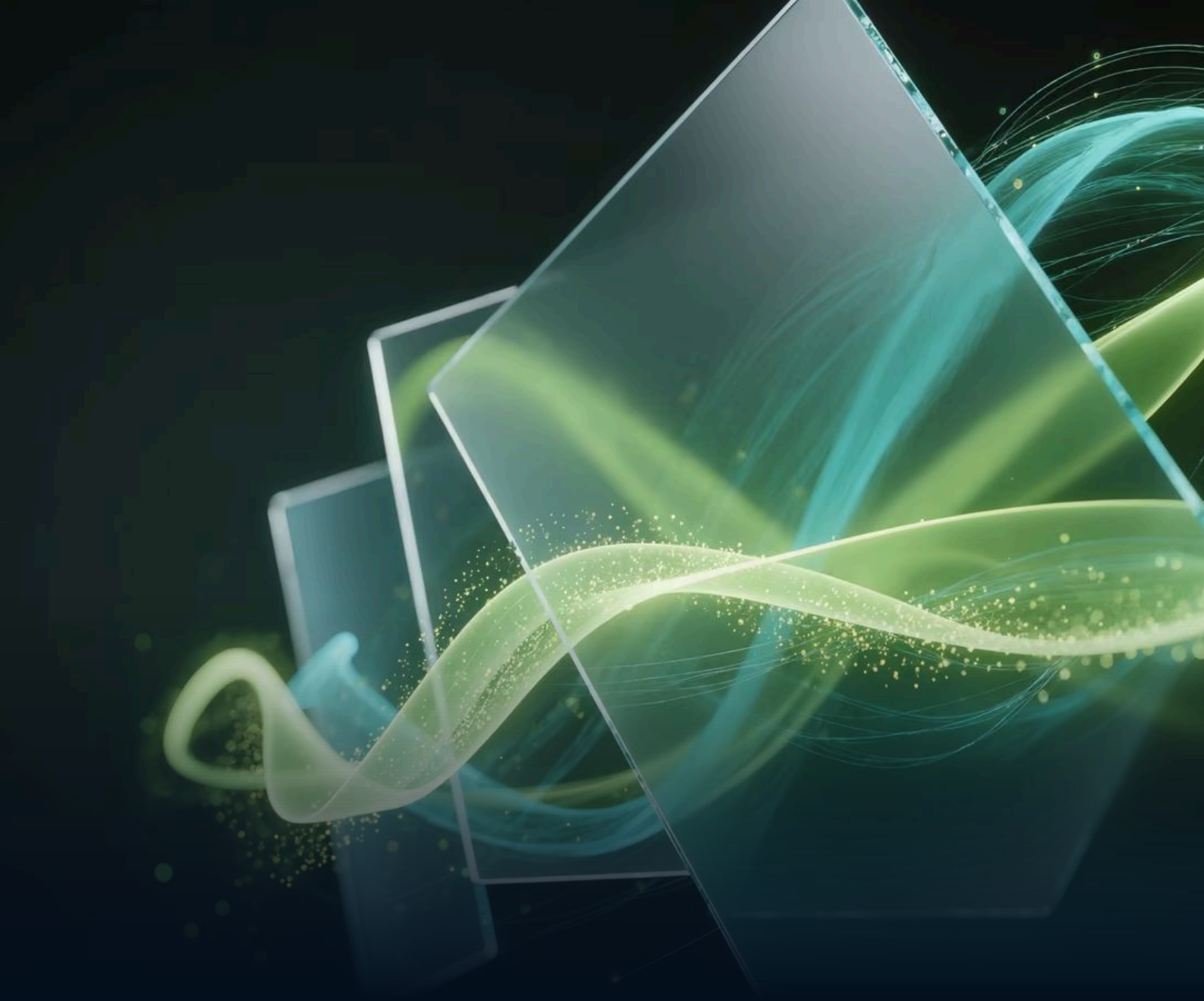
Son certificados profesionales emitidos por Reactor Courses que acreditan horas y competencias. No son un grado académico ni una certificación oficial de Microsoft, AWS, OpenAI o Anthropic. Lo que realmente te va a servir en una entrevista es el repositorio y el sistema desplegado.

El temario menciona productos que cambian rápido. ¿Cómo lo manejan?

Microsoft Foundry, Bedrock AgentCore y los SDK de agentes se mueven cada trimestre. Revalidamos el temario antes de que empiece cada grupo y tienes 12 meses de actualizaciones del material. Por eso el programa enseña primero los conceptos y después el producto: lo que aprendes sobrevive al cambio de versión.

¿Qué pasa si no termino el proyecto final?

Recibes los certificados de las etapas que sí completaste. Por eso la ruta está dividida en cuatro credenciales acumulativas: aunque pauses en la semana 10, no te vas con las manos vacías.



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/agentic-ai-application-engineering.

`reactorcourses.com/programas/agentic-ai-application-engineering`

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.