Reactor
COURSES

BROCHURE DEL PROGRAMA



ARQUITECTURA E INGENIERÍA · DISEÑO Y ESTRATEGIA DE ARQUITECTURA

Solution Architecture Engineering

Diseño, decisión y defensa

Convierte un requerimiento de negocio en una arquitectura de solución dimensionada, costeada, documentada y capaz de sostenerse frente a un comité de revisión.

Lo que vas a lograr Aprendes a convertir un requerimiento de negocio en una arquitectura técnica documentada y costeada, y a defenderla ante un comité.

- C4 Model
- ADR y trade-offs
- Atributos de calidad
- Contratos e integración
- TCO y dimensionamiento
- Architecture Review Board

---

horas en vivo	sesiones	semanas	laboratorios con entregable
72	24	12	24

INICIO
Por confirmar

PRECIO
US$360 pago único · USD

MODALIDAD
Clases en vivo por Zoom · Avanzado

reactorcourses.com/programas/solution-architecture-engineering

EL PROGRAMA

Diseñar la solución y sostener la decisión son dos trabajos distintos. Aquí se hacen los dos.

Convierte un requerimiento de negocio en una arquitectura de solución dimensionada, costeada, documentada y capaz de sostenerse frente a un comité de revisión. Un programa de **72 horas** para ingenieros con oficio que ahora tienen que decidir cómo se construye el sistema, dejarlo por escrito y responder por esa decisión.

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	Lunes y miércoles, 19:00–22:00 (GMT-5)
Precio	US$360 · pago único		

Hay arquitecturas impecables que el equipo no puede construir, y equipos que construyen bien pero no saben explicar por qué eligieron ese camino. El arquitecto de soluciones vive entre esas dos cosas: entrega un diseño ejecutable y lo sostiene cuando alguien lo cuestiona con el presupuesto en la mano.

01

Diseñar la solución

Del problema de negocio al diseño ejecutable

Traducir un requerimiento ambiguo en drivers explícitos, convertirlos en escenarios de calidad medibles y llevarlos a una estructura: contenedores, datos, contratos y despliegue. El diseño se modela, se dimensiona y se documenta para que otro equipo pueda construirlo sin adivinar.

- ✓ Drivers, restricciones, supuestos y casos de uso significativos
- ✓ Atributos de calidad convertidos en escenarios con número de aceptación
- ✓ Modelado C4, vistas dinámicas y vista de despliegue
- ✓ Diseño de datos, contratos síncronos y asíncronos, dimensionamiento

02

Sostener la decisión

Trade-offs, economía y gobierno técnico

Toda decisión de arquitectura es una renuncia. Aquí se aprende a hacer explícito qué se gana, qué se pierde y cuánto cuesta: opciones comparadas con criterios, costos proyectados, riesgos priorizados y una defensa que resiste las preguntas incómodas de un comité.

- ✓ ADR con opciones evaluadas, criterio de decisión y consecuencia
- ✓ TCO, costo unitario y sensibilidad de los supuestos económicos
- ✓ Evaluación por escenarios: puntos de sensibilidad, riesgos y no-riesgos
- ✓ Deuda arquitectónica aceptada, con dueño y fecha de vencimiento

> Una arquitectura no se juzga por lo elegante que se ve en el diagrama, sino por los atributos de calidad que sostiene bajo las restricciones reales de plazo, presupuesto y equipo.

PARA QUIÉN ES

¿Es el programa indicado para ti?

Está pensado para desarrolladores con experiencia que quieren pasar de construir software a decidir cómo se construye.

Es para ti si

- ✓ Desarrolladores y tech leads con tres años o más de oficio que hoy deciden el diseño, no solo lo implementan.
- ✓ Ingenieros que ya hacen de arquitecto sin el título: los llaman cuando hay que elegir entre dos caminos caros.
- ✓ Arquitectos de soluciones en ejercicio que diseñan bien pero pierden la discusión frente a finanzas o a un comité.
- ✓ Consultores y preventa técnica que deben dimensionar, costear y sustentar una propuesta ante un cliente.
- ✓ Ingenieros de plataforma, datos o integración que necesitan el marco de decisión que su especialidad no les dio.

Te conviene otro programa

- – Personas sin experiencia construyendo software. Este programa no enseña a programar ni a operar infraestructura.
- – Quien busque un curso de diagramas: aquí un diagrama sin decisión escrita ni número que lo sostenga se devuelve en la revisión.
- – Quien quiera preparación para un examen de certificación. Este programa no prepara ni acredita para ninguna certificación de terceros.
- – Quien espere escribir código en cada clase. Se lee y se critica código, pero los entregables son artefactos de arquitectura.
- – Quien no pueda dedicar entre 9 y 11 horas por semana durante 12 semanas.
- – Quien necesite la receta cerrada de un proveedor. Aquí se compara y se decide; el producto se elige al final del análisis, no al principio.

EL RESULTADO

Terminas con un dossier de arquitectura completo y defendido, no con un certificado de asistencia.

El proyecto final es el paquete que un comité de arquitectura te pediría en una empresa real. Es exactamente el material con el que se gana una entrevista de arquitecto o se aprueba una propuesta interna.

01 **Documento de arquitectura** Un SAD vivo con alcance, drivers, restricciones, vistas, decisiones y riesgos.

02 **Drivers y restricciones** Stakeholders mapeados, supuestos declarados y casos de uso significativos priorizados.

03 **Escenarios de calidad** Atributos convertidos en escenarios con estímulo, respuesta y número de aceptación.

04 **Vistas C4** Contexto, Contenedores y Componentes, más vistas dinámicas y de despliegue.

05 **Diez ADR como mínimo** Opciones evaluadas, criterio de decisión, decisión tomada y consecuencia asumida.

06 **Diseño de datos** Modelo lógico, propiedad del dato, consistencia y motor elegido con justificación.

07 **Contratos de integración** OpenAPI para lo síncrono, contrato de eventos para lo asíncrono, versionado y compatibilidad.

08 **Dimensionamiento** Modelo de carga, cálculo de capacidad y presupuesto de latencia del camino crítico.

09 **Modelo de costos** TCO a tres años, costo unitario por transacción y sensibilidad de los supuestos.

10 **Plan de entrega** Fases, dependencias, ruta de migración y criterios de corte y de reversión.

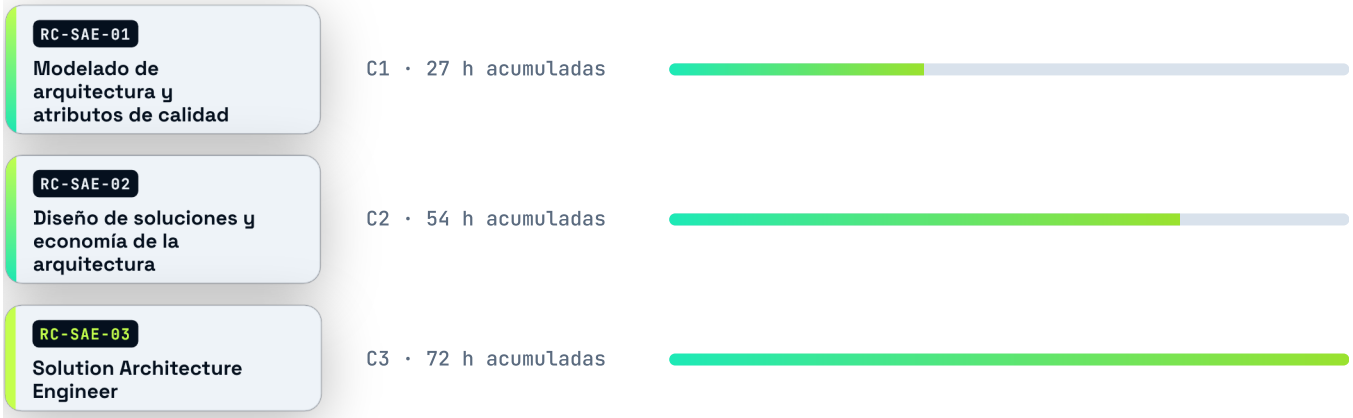
11 **Riesgos y trade-offs** Puntos de sensibilidad, riesgos priorizados, mitigaciones y deuda aceptada con vencimiento.

12 **Defensa ante comité** Architecture Review Board: sustentas el diseño y respondes preguntas hostiles en vivo.

RUTA DE CERTIFICACIÓN

3 credenciales progresivas, no una al final.

Cada etapa cierra un bloque completo de competencias. Si algo te obliga a pausar en la semana 9, ya tienes dos credenciales defendibles en tu CV.



CERTIFICACIÓN INTEGRAL

Solution Architecture Engineering — Diseño, decisión y defensa (72 horas)

Requiere entregar el dossier de arquitectura y aprobar la defensa ante el comité. 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 OFICIO DEL ARQUITECTO DE SOLUCIONES — 9 h · 3 sesiones

01 Qué decide un arquitecto de soluciones y qué no

alcance real del rol · arquitecto de soluciones vs tech lead vs arquitecto empresarial · qué hace significativa a una decisión · costo del cambio en el tiempo · entregables por los que responde el arquitecto · ciclo de vida de una solución · la arquitectura como conjunto de restricciones deliberadas · quién firma la decisión y quién responde por ella

LABORATORIO: Delimitar el alcance de decisión del caso del programa y elegir el caso de negocio propio

02 Del problema de negocio a los drivers de arquitectura

mapa de stakeholders y sus intereses en conflicto · objetivos de negocio traducibles a diseño · restricciones técnicas, de organización y de plazo · supuestos declarados y supuestos ocultos · casos de uso significativos para la arquitectura · qué requisito cambia la estructura y cuál no · alcance y no-alcance por escrito · cómo se conduce la entrevista al stakeholder

LABORATORIO: Mapa de stakeholders y tabla de drivers, restricciones y supuestos del caso

03 Atributos de calidad medibles

ISO/IEC 25010 como vocabulario común · atributo de calidad vs requisito funcional · escenario de calidad de seis partes · fuente, estímulo, artefacto, entorno, respuesta y medida · priorización por impacto de negocio y por riesgo técnico · tácticas de arquitectura asociadas a cada atributo · convertir «tiene que ser rápido» en un número de aceptación · el atributo que nadie pide y siempre termina cobrando

LABORATORIO: Ocho escenarios de calidad priorizados con su medida de aceptación — compuerta Drivers Ready

MÓDULO 2 MODELADO Y DOCUMENTACIÓN DE LA ARQUITECTURA — 9 h · 3 sesiones

04 C4: contexto y contenedores

los cuatro niveles del C4 Model · diagrama de contexto del sistema · personas, sistema en foco y sistemas externos · diagrama de contenedores · qué es un contenedor y qué no lo es · notación, leyenda y consistencia visual · el error de mezclar niveles en un mismo diagrama · diagramas como código con Structurizr DSL

LABORATORIO: Diagramas de Contexto y de Contenedores del caso, versionados en el repositorio

05 Vistas de componentes, dinámicas y de despliegue

diagrama de componentes del contenedor crítico · vista dinámica de un caso de uso de negocio · vista de despliegue y nodos de infraestructura · el modelo 4+1 y qué vistas se usan de verdad · trazabilidad entre vistas de un mismo modelo · cuánto detalle es demasiado detalle · mantener el modelo vivo sin que se vuelva una carga · generar todos los diagramas desde un modelo único

LABORATORIO: Vista de componentes del contenedor crítico y vista dinámica del camino de negocio principal

06 SAD y ADR: decisiones que sobreviven a su autor

estructura de un Software Architecture Document · documentación mínima suficiente · formato MADR para registros de decisión · contexto, opciones evaluadas, criterios, decisión y consecuencias · decisión reversible frente a decisión irreversible · cómo se supersede un ADR sin borrar la historia · el resumen de una página para dirección · el SAD versionado junto al código

LABORATORIO: Esqueleto del SAD publicado en el repositorio y los tres primeros ADR del caso

MÓDULO 3 ESTRUCTURA, DATOS Y CONTRATOS — 9 h · 3 sesiones

07 Estilos arquitectónicos y criterios de descomposición

monolito modular · capas y arquitectura hexagonal · servicios distribuidos · orientación a eventos · procesamiento por lotes · descomposición por capacidad de negocio · acoplamiento y cohesión a escala de sistema · el costo organizacional de cada estilo · cuándo el monolito es la respuesta correcta

LABORATORIO: Comparar dos estilos candidatos con criterios explícitos y escribir el ADR de la decisión

08 **Diseño de datos de la solución**

modelo lógico y propiedad del dato · límites transaccionales del negocio · consistencia fuerte y consistencia eventual · elección de motor: relacional, documental, clave-valor, búsqueda, analítico · carga operacional frente a carga analítica · duplicación deliberada y sincronización · retención, ciclo de vida y datos personales · la migración de datos como restricción de diseño

LABORATORIO: Modelo lógico de datos, matriz de propiedad del dato y ADR de elección de motor

09 **Contratos e integración**

integración síncrona sobre HTTP · mensajería asíncrona y eventos de negocio · archivos y procesos por lotes · diseñar el contrato antes que la implementación · OpenAPI y AsyncAPI · idempotencia y entrega al menos una vez · versionado y compatibilidad hacia atrás · capa anticorrupción frente a sistemas heredados

LABORATORIO: Contrato OpenAPI de la API principal y contrato del evento de negocio crítico

MÓDULO 4 ATRIBUTOS DE CALIDAD EN EL DISEÑO — 9 h · 3 sesiones

10 **Disponibilidad y resiliencia como decisión de diseño**

SLA, SLO y SLI · presupuesto de error · modos de falla y análisis de impacto · puntos únicos de falla · redundancia por zonas y por regiones · degradación funcional y modo limitado · reintentos, timeouts y cortacircuitos como decisión estructural · RTO y RPO del caso

LABORATORIO: SLO del servicio crítico, análisis de modos de falla y decisión de redundancia con su costo

11 **Rendimiento, capacidad y dimensionamiento**

modelo de carga: usuarios, transacciones, pico y estacionalidad · latencia por percentiles · presupuesto de latencia del camino crítico · estimación de orden de magnitud · cálculo de cómputo, almacenamiento y ancho de banda · caché: dónde ayuda y dónde esconde el problema · escalado vertical, horizontal y sus límites reales · cómo se valida una hipótesis de rendimiento antes de construir

LABORATORIO: Modelo de carga y dimensionamiento del caso, con hoja de cálculo y supuestos declarados

12 **Seguridad y cumplimiento en el diseño**

identidad de usuarios e identidad de servicios · autenticación y autorización como decisión estructural · clasificación de datos y tratamiento de datos personales · cifrado en tránsito y en reposo · segmentación de red y superficie expuesta · gestión de secretos · trazas de auditoría y no repudio · requisitos regulatorios como restricción de arquitectura

LABORATORIO: Modelo de amenazas de alto nivel y decisiones de seguridad en ADR — compuerta Design Ready

MÓDULO 5 ECONOMÍA DE LA SOLUCIÓN — 9 h · 3 sesiones

13 **Costear una arquitectura**

TCO a tres años · CAPEX y OPEX · costo de infraestructura, licencias, personas y operación · unidad de costo del negocio · costo por transacción y costo por usuario · análisis de sensibilidad de los supuestos · el costo de la arquitectura que no se eligió · presentar el número a alguien que no es técnico

LABORATORIO: Modelo de costos del caso: TCO a tres años, costo unitario y sensibilidad de dos supuestos

14 **Construir, comprar o contratar como servicio**

diferenciación de negocio como criterio de decisión · costo total de comprar frente a construir · modelos de licenciamiento y de precio · dependencia de proveedor y costo de salida · riesgo de proveedor y continuidad del servicio · la integración como costo oculto · matriz de decisión con criterios ponderados · la decisión reversible se toma rápido y se documenta igual

LABORATORIO: Matriz de decisión construir/comprar/servicio de un componente del caso y su ADR

15 **Estimación y plan de entrega**

descomposición del alcance en incrementos con valor · dependencias y ruta crítica · incertidumbre expresada en rangos, no en fechas únicas · primer incremento entregable y qué demuestra · tamaño y forma del equipo necesario · la ley de Conway y la estructura organizacional · hitos de arquitectura dentro del plan · qué se compromete y qué solo se estima

LABORATORIO: Plan de entrega en fases con dependencias, rangos de esfuerzo e hitos de arquitectura

MÓDULO 6 EVOLUCIÓN DE SISTEMAS EXISTENTES — 9 h · 3 sesiones

16 **Diagnóstico de un sistema en producción**

inventario de aplicaciones e integraciones · leer código existente para detectar acoplamiento · deuda técnica y deuda arquitectónica · mapa de calor por riesgo y por frecuencia de cambio · dependencias ocultas y datos compartidos · medir antes de opinar · entrevistar al equipo que lo mantiene · qué se conserva, y el argumento para conservarlo

LABORATORIO: Diagnóstico de un sistema heredado con mapa de calor y riesgos priorizados

17 **Estrategias de evolución**

mantener, encapsular, refactorizar, reemplazar, retirar · reemplazo incremental por capacidades de negocio · fachada e interceptación de tráfico · coexistencia y sincronización de datos entre lo viejo y lo nuevo · doble escritura y sus trampas · el patrón de estrangulamiento y los casos en que falla · reescritura completa: cuándo es honesto proponerla · criterios para detener una migración a mitad de camino

LABORATORIO: Ruta de evolución del sistema heredado por capacidades, con su ADR

18 **Plan de migración y corte**

fases, ventanas de corte y ensayo previo · migración de datos y reconciliación · corte total, por segmentos o en paralelo · criterios de éxito medibles · criterios de aborto y procedimiento de reversión · convivencia temporal de dos sistemas · comunicación a usuarios y a negocio · el costo de la ventana de riesgo

LABORATORIO: Plan de migración y corte con criterios de éxito, de aborto y procedimiento de reversión

MÓDULO 7 EVALUACIÓN, RIESGO Y COMUNICACIÓN — 9 h · 3 sesiones

19 **Evaluar una arquitectura por escenarios**

por qué una arquitectura se evalúa antes de construirla · árbol de utilidad de atributos de calidad · escenarios de evaluación y su priorización · punto de sensibilidad · punto de trade-off · riesgo y no-riesgo · conducir una sesión de evaluación con stakeholders · qué se hace con los hallazgos y quién los cierra

LABORATORIO: Evaluar por escenarios la arquitectura de otro equipo y entregar el informe de hallazgos

20 **Riesgo y deuda arquitectónica**

identificación y clasificación de riesgos · probabilidad, impacto y exposición · mitigar, transferir, aceptar o evitar · riesgo residual declarado por escrito · deuda arquitectónica aceptada a conciencia · dueño y fecha de vencimiento de cada deuda · indicadores tempranos de que la arquitectura se está rompiendo · el registro de riesgos como entregable vivo

LABORATORIO: Registro de riesgos del caso y tabla de deuda aceptada con dueño y vencimiento

21 **Comunicar y defender una arquitectura**

audiencia técnica, audiencia ejecutiva y audiencia financiera · el argumento completo en una página · presentar un trade-off sin sonar evasivo · responder «¿por qué no lo hiciste más simple?» · negociar una restricción en lugar de aceptarla · qué busca realmente un comité de arquitectura · decir que no con datos sobre la mesa · qué hacer cuando la decisión te la cambian desde arriba

LABORATORIO: Documento de una página y presentación ejecutiva del caso, ensayados en clase

MÓDULO 8 **DOSSIER DE ARQUITECTURA Y DEFENSA** 9 h · 3 sesiones

22 **Taller de consolidación del dossier**

integrar vistas, decisiones y números en un solo documento · coherencia entre el SAD y cada ADR · trazabilidad de driver a decisión y de decisión a escenario · los huecos que aparecen siempre · supuestos que quedaron sin declarar · lista de verificación de completitud · control de versiones del dossier · qué se entrega y en qué formato

LABORATORIO: Dossier consolidado y verificado contra la lista de completitud

23 **Revisión cruzada y ensayo de defensa**

revisión por pares con los criterios de la evaluación por escenarios · las preguntas hostiles más frecuentes · detectar decisiones sin evidencia detrás · corregir sin romper la coherencia del diseño · priorizar los hallazgos por impacto · estructura y tiempo de una defensa · qué se responde en el momento y qué se anota para después · compuerta Review Ready

LABORATORIO: Informe de revisión cruzada recibido y emitido, y plan de corrección del dossier

24 **Architecture Review Board: defensa final**

¿qué driver justifica esta estructura? · ¿qué escenario de calidad falla primero? · ¿de dónde salió ese número? · ¿cuál es el punto de trade-off de la decisión central? · ¿qué pasa si el volumen crece diez veces? · ¿por qué construir en lugar de comprar? · ¿cuál es el plan de reversión? · ¿qué deuda estás aceptando y hasta cuándo?

LABORATORIO: Defensa individual del dossier ante el comité y acta de decisión

HERRAMIENTAS Y MARCOS

Todo lo que vas a usar para diseñar, medir y sustentar.

Modelado

C4 Model · Structurizr DSL · PlantUML · Mermaid · Diagramas como código

Decisión documentada

SAD · ADR (formato MADR) · RFC técnico · Documento de una página

Contratos

OpenAPI · AsyncAPI · JSON Schema · Versionado y compatibilidad · Contract testing

Atributos de calidad

ISO/IEC 25010 · Escenarios de calidad · Tácticas de arquitectura · SLO · Presupuesto de error

Dimensionamiento y costo

Modelo de carga · Estimación de orden de magnitud · Calculadoras de precios de nube · TCO · Costo unitario

Evaluación y riesgo

Evaluación por escenarios · Puntos de sensibilidad · Matriz de riesgo · Deuda arquitectónica

Cuerpos de conocimiento

Well-Architected · TOGAF ADM · DDD estratégico · Twelve-Factor App

Herramientas de trabajo

Git · Markdown · Hoja de cálculo · Pizarra colaborativa · Docker para Structurizr Lite

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, taller permanente y una decisión escrita en cada entrega.

01 **Clases en vivo por Zoom** 24 sesiones de 3 horas, 2 veces por semana. Cámara y micro abiertos: la arquitectura se aprende discutiendo, no mirando.

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% taller** La mayor parte del tiempo diseñas, comparas opciones y escribes la decisión. Cada sesión cierra con un entregable que entra al dossier.

04 **Un caso que dura 12 semanas** Trabajas un caso de negocio completo desde la sesión 01. Cada módulo le agrega una capa y el acumulado es tu dossier final.

05 **Revisión cruzada entre equipos** Tu diseño lo critica otro equipo antes de que lo critique el comité. Recibir la crítica sin ponerte a la defensiva es parte del oficio.

06 **Tres compuertas de calidad** Drivers Ready, Design Ready y Review Ready. No se avanza con supuestos sin declarar ni con decisiones sin un número detrás.

Horarios · Primer grupo

Lunes	19:00 - 22:00
Miércoles	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 asume que ya construiste software en serio y que puedes leer un diseño sin que te lo traduzcan.

Experiencia

- ✓ Tres años o más construyendo software
- ✓ Haber participado en un sistema en producción
- ✓ Haber vivido al menos un problema serio de rendimiento, datos o integración

Desarrollo

- ✓ Leer código de al menos un lenguaje
- ✓ Capas, dependencias y acoplamiento
- ✓ Pruebas automatizadas a nivel conceptual
- ✓ Control de versiones en equipo

Datos

- ✓ Modelo relacional y SQL
- ✓ Índices y transacciones
- ✓ Nociones de motores no relacionales
- ✓ Diferencia entre carga transaccional y analítica

Integración

- ✓ HTTP y verbos REST
- ✓ JSON y contratos de API
- ✓ Nociones de mensajería
- ✓ Procesos batch y archivos

Infraestructura

- ✓ Cómputo, almacenamiento y red
- ✓ Balanceo y DNS a nivel conceptual
- ✓ Identidad, secretos y logging
- ✓ Qué es un entorno y qué es un despliegue

Herramientas

- ✓ Git a nivel de rama y pull request
- ✓ Markdown
- ✓ Hoja de cálculo para hacer números
- ✓ Docker para levantar una herramienta local

PRECIO

72 horas para dejar de improvisar la decisión técnica.

US$360

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
- ✓ El caso de negocio del programa y las plantillas de SAD, ADR y escenarios de calidad
- ✓ Revisión de entregables en las 3 compuertas de calidad
- ✓ Revisión cruzada entre equipos en cada módulo
- ✓ Dossier de arquitectura con acompañamiento y defensa ante comité
- ✓ Los 3 certificados progresivos + la certificación integral
- ✓ Canal privado del grupo para dudas entre sesiones
- ✓ Actualizaciones del material por 12 meses

No incluye

- — Licencias de herramientas comerciales de modelado: el programa se resuelve entero con alternativas gratuitas o de código abierto
- — Exámenes ni certificaciones de terceros: este programa no prepara para ninguno
- — Consultoría sobre el sistema de tu empresa fuera del horario de clase

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, así que puedes recuperar una clase perdida o repasar una discusión de diseño.

¿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 Lunes y miércoles 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 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 discusión en vivo, que en un programa de arquitectura es la mitad del valor. Escríbenos antes de comprar si tu diferencia horaria supera las tres horas.

¿Necesito saber programar?

Necesitas haber programado en serio, aunque hoy ya no escribas código todos los días. El programa no enseña a programar y asume que puedes leer código, entender un modelo de datos y discutir una integración sin traducción. Revisa la sección de requisitos con honestidad antes de inscribirte.

¿Este programa prepara para alguna certificación?

No. No prepara ni acredita para ninguna certificación de AWS, Microsoft, Google, The Open Group ni ninguna otra organización. El contenido se apoya en cuerpos de conocimiento públicos como los marcos Well-Architected, el ADM de TOGAF y el diseño estratégico de DDD, pero Reactor Courses no tiene relación, aval ni acreditación de esas organizaciones, y los certificados que emitimos son propios.

¿Se trabaja sobre una nube en particular?

No se casa con ninguna. Los ejercicios de dimensionamiento y costo se hacen con las calculadoras públicas de los proveedores grandes y se comparan entre sí, porque comparar es justamente la competencia que se entrena. Si tu empresa ya eligió proveedor, puedes hacer todo el caso sobre ese y el resultado sigue siendo válido.

¿Voy a escribir código?

Poco, y nunca como entregable principal. Se lee código para evaluar acoplamiento y deuda, se escriben contratos OpenAPI y esquemas de eventos, y se modela con diagramas como código. Si lo que buscas es un programa de implementación, este no lo es y preferimos decírtelo ahora.

¿Puedo usar un caso de mi empresa como proyecto final?

Sí, y suele ser lo más rentable. La condición es que no traigas información confidencial: se trabaja con el problema y los números anonimizados o aproximados. Si prefieres no exponer nada de tu trabajo, el programa incluye un caso de negocio completo listo para usar.

¿Cuánto tiempo debo dedicarle por semana?

Seis horas de clase más entre tres y cinco horas de taller y entregables. Considera entre 9 y 11 horas semanales durante 12 semanas.

¿Qué costos adicionales tengo?

Ninguno obligatorio. Todo el laboratorio se resuelve con herramientas gratuitas o de código abierto: Structurizr Lite, PlantUML, Mermaid, Git, Markdown, una hoja de cálculo y las calculadoras públicas de precios de nube. No se despliega infraestructura, así que no hay consumo que pagar.

¿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$360. 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?

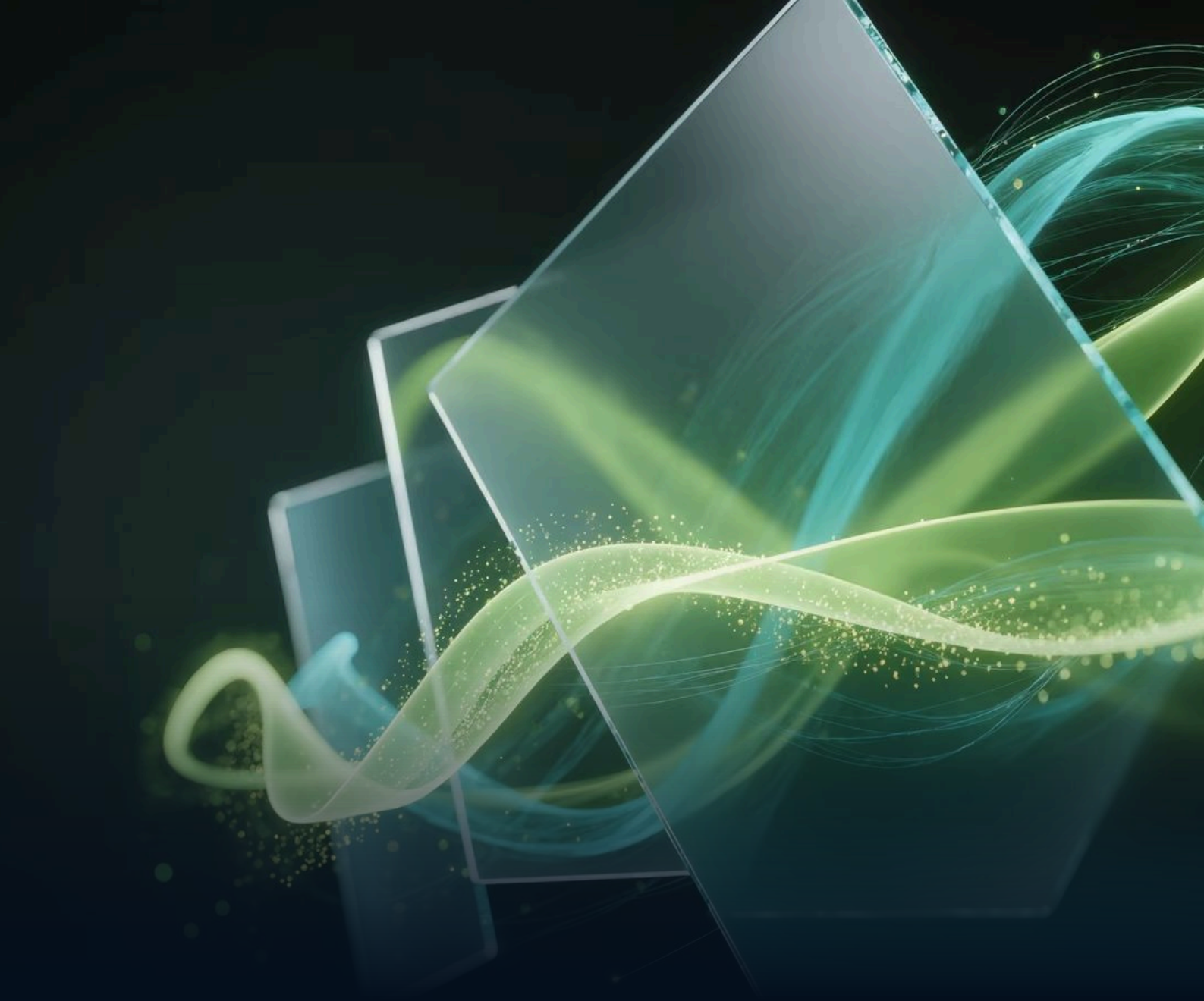
Tienes 14 días desde el inicio del programa para pedir el reembolso completo escribiendo a soporte@reactorcourses.com. Publicamos el temario completo y los requisitos justamente para que decidas informado y nadie llegue a ese punto.

¿Los certificados tienen validez universitaria?

No. Son certificados profesionales emitidos por Reactor Courses que acreditan horas y competencias. No son un grado académico ni una certificación oficial de ninguna organización. Lo que de verdad te va a servir en una entrevista es el dossier de arquitectura y haberlo defendido.

¿Este programa se cruza con otros del catálogo?

Cada programa de Reactor Courses es un producto cerrado: trae sus propios fundamentos, su propio proyecto final y no da por visto nada de otro curso. Dicho eso, el modelado C4 y las decisiones en ADR aparecen también en otros programas de la escuela, tratados desde su propio problema. Aquí son el centro del trabajo y se llevan hasta la evaluación por escenarios, el costeo y la defensa ante comité. Si cursas dos programas vas a volver a ver esos temas, y no hay convalidaciones ni descuentos entre programas: preferimos decirlo antes y no que lo descubras pagando.



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/solution-architecture-engineering.

`reactorcourses.com/programas/solution-architecture-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.