Reactor
COURSES

BROCHURE DEL PROGRAMA



ARQUITECTURA E INGENIERÍA · CONFIABILIDAD Y RENDIMIENTO

SRE & Resilience Architecture

La confiabilidad como decisión de producto

Convierte la confiabilidad en un número acordado con el negocio, opera el servicio contra ese número —alertas, presupuesto de error, incidentes y postmortems— y diseña el sistema para que falle de forma acotada.

Lo que vas a lograr Aprendes a acordar cuánto debe funcionar un servicio, a medirlo, a alertar solo cuando importa y a responder a las caídas con método.

- SLI, SLO y presupuesto de error
- Alertas por tasa de consumo
- OpenTelemetry
- Resiliencia y capacidad
- Gestión de incidentes
- Game day y postmortems

horas en vivo	sesiones	semanas	fallas sin aviso en el game day
72	24	12	3

INICIO
Por confirmar

PRECIO
US$400 pago único · USD

MODALIDAD
Clases en vivo por Zoom · Avanzado

reactorcourses.com/programas/sre-resilience-architecture

EL PROGRAMA

Si no puedes decir qué tan confiable debe ser tu servicio, no puedes saber si lo es.

Convierte la confiabilidad en un número acordado con el negocio, opera el servicio contra ese número — alertas, presupuesto de error, incidentes y postmortems— y diseña el sistema para que falle de forma acotada. Un programa de **72 horas** para ingenieros que hacen guardias y quieren que dejen de ser una lotería, y para equipos que tienen que introducir SLO en una organización que todavía no los usa.

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

La mayoría de los equipos operan con alertas que nadie eligió, tableros que nadie mira y la sensación de que el próximo incidente es cuestión de suerte. Este programa convierte la confiabilidad en un número acordado con el negocio y enseña a operar contra ese número: qué medir, cuándo despertar a alguien, cuánto riesgo se puede tomar y qué se aprende de cada incidente.

01

Medir lo que el usuario sufre

SLI, SLO, presupuesto de error y observabilidad

Indicadores que bajan cuando el usuario sufre y no cuando el error es suyo, objetivos elegidos con historial y no con deseo, una política de presupuesto que se cumple, y alertas por tasa de consumo que avisan antes del daño sin despertar a nadie por una falla breve.

- SLI de disponibilidad, latencia, calidad y frescura
- SLO negociados con producto y escritos como código
- Alertas de varias ventanas generadas desde el SLO
- Instrumentación con OpenTelemetry y diagnóstico por ejemplares

02

Fallar acotado y aprender de cada falla

Resiliencia, capacidad, cambio seguro e incidentes

Patrones elegidos por el presupuesto que ahorran, degradación por criticidad, pruebas de carga que no mienten, canarios que se revierten solos, y la parte que más falla en la realidad: coordinar un incidente con personas, escribir un postmortem que cambia algo y sostener guardias que no queman al equipo.

- Patrones de resiliencia medidos antes y después bajo la misma falla
- Capacidad con margen para perder una zona
- Entrega progresiva con puerta según el presupuesto
- Incidentes, postmortems sin culpables y guardias sostenibles

> Un SLO de 99,99 % que nadie negoció no es un objetivo: es una promesa que el equipo de guardia va a pagar con sueño.

PARA QUIÉN ES

¿Es el programa indicado para ti?

Está pensado para quienes operan servicios en producción y quieren que fallen menos y se recuperen más rápido.

Es para ti si

- ✓ Ingenieros que hacen guardias y quieren que dejen de ser una lotería.
- ✓ Equipos de plataforma o DevOps que tienen que introducir SLO en su organización.
- ✓ Líderes técnicos responsables de incidentes y postmortems.
- ✓ Desarrolladores que operan lo que construyen y quieren hacerlo con método.
- ✓ Responsables de producto técnico que negocian confiabilidad frente a nuevas funciones.

Te conviene otro programa

- – Quien no haya operado un servicio en producción: el programa trabaja sobre incidentes reales, incluidos los tuyos.
- – Quien busque un curso de una herramienta de monitoreo: las herramientas son medios y se pueden cambiar.
- – Quien espere teoría de consistencia y consenso distribuido: aquí no se demuestra que un algoritmo es correcto.
- – Quien ya cursó un programa de sistemas distribuidos: cerca de un tercio de este programa —tiempos de espera, cortacircuitos, inyección de fallas, capacidad, recuperación, canarios e incidentes— trata problemas parecidos desde el servicio en producción. Lee en las preguntas frecuentes la que trata ese caso.
- – Quien espere sumar el título de SRE con un certificado: el oficio se construye operando, y este programa da método y práctica, no un cargo.
- – Quien no tenga un equipo con 16 GB de memoria o no pueda dedicar entre 10 y 12 horas por semana durante 12 semanas.

EL RESULTADO

Terminas con un servicio operado contra sus SLO, probado con fallas que no esperabas.

Cada laboratorio deja una pieza del servicio de reservas del caso con la prueba de que funciona. Juntas forman el proyecto que defiendes al final, después de un game day sin aviso.

01 **SLI que miden al usuario** Cuatro indicadores implementados que bajan cuando el usuario sufre y no cuando el error es del cliente.

02 **SLO defendidos con cifras** Objetivos elegidos a partir del historial, escritos como código y defendidos ante el grupo.

03 **Política de presupuesto aplicada** Disparadores, responsables y excepciones, aplicados a tres meses simulados de incidentes.

04 **Instrumentación de punta a punta** Tres servicios instrumentados con OpenTelemetry, con la cardinalidad de métricas bajo el límite fijado.

05 **Alertas que avisan a tiempo** Alertas de varias ventanas que disparan antes de gastar el 10 % del presupuesto y callan ante fallas breves.

06 **Diagnóstico con reloj** Tres fallas desconocidas resueltas desde el aviso hasta la causa, con el tiempo de diagnóstico registrado.

07 **Resiliencia medida** Patrones aplicados y degradación por criticidad, con el presupuesto ahorrado medido bajo la misma falla.

08 **Capacidad y carga** Prueba de carga de modelo abierto y plan de capacidad con margen para perder una zona.

09 **Recuperación y caos** Simulacro de recuperación medido contra sus objetivos y dos experimentos de caos con criterio de aborto.

10 **Cambio seguro** Canario que se revierte solo por SLI y una puerta que bloquea despliegues con el presupuesto agotado.

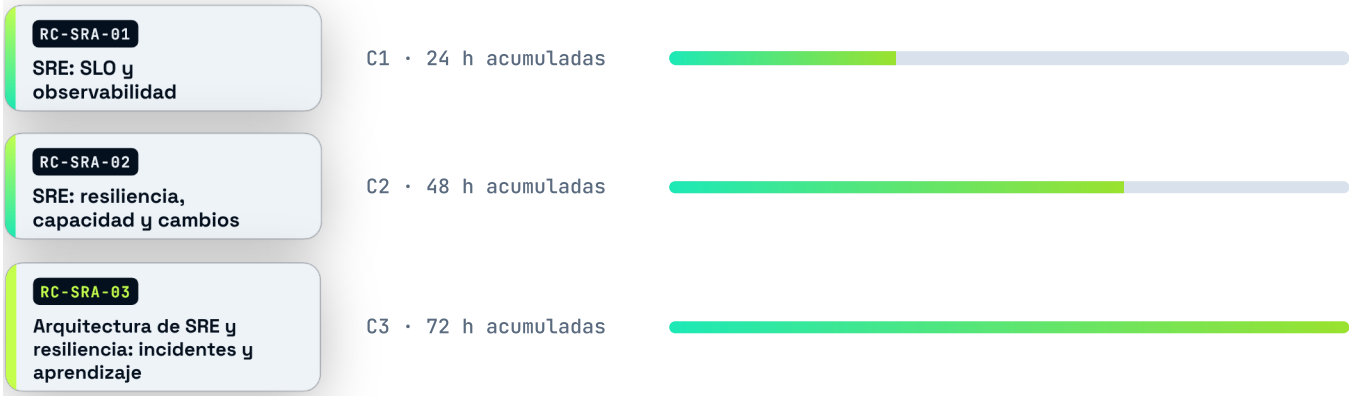
11 **Incidente y postmortem** Incidente coordinado con roles y comunicación, y postmortem con acciones verificables revisado en grupo.

12 **Guardia y trabajo repetitivo** Carga de guardia reducida a la mitad y una automatización con las horas recuperadas calculadas.

RUTA DE CERTIFICACIÓN

3 credenciales progresivas, no una al final.

Cada etapa cierra un bloque completo: primero medir y alertar, después fallar acotado y cambiar con seguridad, y al final gestionar incidentes y aprender de ellos. Son credenciales de Reactor Courses, no certificaciones de terceros.



CERTIFICACIÓN INTEGRAL

SRE & Resilience Architecture — La confiabilidad como decisión de producto (72 horas)

Requiere entregar el servicio con sus SLO y alertas, la respuesta al game day y el postmortem con las acciones verificadas. 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 LA CONFIABILIDAD COMO NÚMERO — 12 h · 4 sesiones

01 El servicio en producción y sus usuarios

el caso del programa: reservas de una red de clínicas con pagos de un proveedor externo · recorridos críticos de usuario y qué significa que funcionen · confiabilidad como atributo que el usuario percibe, no como uptime del servidor · incidentes reales de los alumnos como materia prima · costo de la confiabilidad: cada nueve adicional y quién lo paga · el entorno local: servicios en contenedores, colector de telemetría y almacenamiento de métricas · qué se mide hoy en el caso y qué nadie mira

LABORATORIO: Levantar con Docker Compose el servicio de reservas con su telemetría, inyectar una falla de la dependencia de pagos y registrar qué vio el usuario, qué mostraban los tableros existentes y cuánto tardó alguien en enterarse

02 **Indicadores de nivel de servicio que miden al usuario**

SLI de disponibilidad, latencia, calidad y frescura · medir en el balanceador, en el servicio o desde el cliente · eventos buenos sobre eventos válidos · qué excluir: errores del cliente, tráfico de prueba y robots · latencia por percentil y por umbral · SLI de procesos asincrónicos y de colas · especificación frente a implementación de un SLI

LABORATORIO: Definir cuatro SLI para dos recorridos del caso, implementarlos como consultas sobre las métricas del servicio y demostrar con tráfico sintético que cada uno baja cuando el usuario sufre y no baja cuando el error es del cliente

03 **Elegir el objetivo y negociarlo**

por qué 99,99 % no es un valor por defecto · ventana de cumplimiento: móvil o de calendario · partir del historial y no del deseo · SLO, SLA y la penalidad que nadie leyó · negociar con producto: costo de cada nueve y qué se deja de construir · objetivos por recorrido y por tipo de cliente · SLO como código y su revisión periódica

LABORATORIO: Elegir los SLO del caso a partir de 30 días de historial sintético, escribirlos como código con OpenSLO y defender ante el grupo, con cifras, por qué no son más altos

04 **Presupuesto de error y política**

presupuesto de error como cantidad de fallas permitidas · política de presupuesto: qué pasa cuando se agota · congelamiento de cambios y sus excepciones · quién firma la política y quién la hace cumplir · presupuesto consumido por dependencias y por despliegues · informe de confiabilidad para el negocio · cuando el presupuesto sobra: gastar en velocidad

LABORATORIO: Redactar la política de presupuesto de error del caso con disparadores, responsables y excepciones, y aplicarla a tres meses simulados de incidentes decidiendo en cada corte si se congela, se sigue o se acelera

MÓDULO 2 OBSERVABILIDAD PARA OPERAR — 12 h · 4 sesiones

05 **Instrumentación con OpenTelemetry**

trazas, métricas y logs: estabilidad actual de cada señal en OpenTelemetry · instrumentación automática y manual · convenciones semánticas y nombres que se entienden · propagación de contexto entre servicios y colas · el colector: recepción, procesamiento y exportación · cardinalidad de métricas y su costo · muestreo en la cabeza y en la cola

LABORATORIO: Instrumentar los tres servicios del caso con OpenTelemetry, exportar a Prometheus, Jaeger y Loki por el colector, y demostrar que una petición lenta se sigue de punta a punta sin que la cardinalidad de métricas pase el límite fijado

06 **Alertas por tasa de consumo del presupuesto**

por qué alertar por síntoma y no por causa · tasa de consumo del presupuesto y su lectura · alertas de varias ventanas y varias tasas del Workbook de SRE · ventana corta de un doceavo de la larga · página frente a ticket · reglas generadas desde el SLO con Sloth · precisión, exhaustividad y tiempo de detección de una alerta

LABORATORIO: Generar con Sloth las alertas de varias ventanas, enrutarlas con Alertmanager para los SLO del caso y demostrar con tres fallas inyectadas —brusca, lenta e intermitente— que disparan antes de agotar el 10 % del presupuesto y que una falla breve sin impacto no despierta a nadie

07 **Del aviso a la causa**

tableros por SLO y no por servidor · ejemplares: de la métrica a la traza que la explica · correlación de traza, métrica y log por identificador · hipótesis y descarte durante un diagnóstico · diagnóstico de latencia de cola · ruido de los tableros y cómo podarlos · tiempo de diagnóstico como métrica del equipo

LABORATORIO: Resolver en parejas, con reloj, tres fallas desconocidas partiendo del aviso de consumo del presupuesto hasta la causa, usando tableros de Grafana, ejemplares y trazas, y registrar el tiempo de diagnóstico de cada una

08 **Dependencias y presupuesto heredado**

la confiabilidad de un servicio no supera la de sus dependencias críticas · SLO de proveedores externos y lo que realmente comprometen · medir a la dependencia desde nuestro lado · presupuesto consumido por terceros y su atribución · dependencia crítica frente a dependencia opcional · conversación con un proveedor que incumple

LABORATORIO: Calcular el techo de confiabilidad del caso a partir de las dependencias medidas, atribuir el presupuesto consumido el último mes entre servicio propio y proveedor de pagos, y proponer qué dependencia pasar a opcional

MÓDULO 3 FALLAR DE FORMA ACOTADA

6 h · 2 sesiones

09 Patrones de resiliencia elegidos por presupuesto

tiempo de espera, reintento con retroceso y dispersión · cortacircuitos y compartimentos · desprendimiento de carga · cada patrón medido por el presupuesto que ahorra · patrones que empeoran el incidente si se configuran mal · configuración en el código, en el proxy o en la malla · el mismo fallo antes y después

LABORATORIO: Aplicar tres patrones de resiliencia al caso y medir, bajo la misma falla de la dependencia de pagos, el presupuesto de error consumido antes y después de cada uno, descartando el que no ahorra presupuesto

10 Degradación elegante por criticidad

niveles de criticidad de cada función del producto · respuesta parcial y valor de último conocido · apagar funciones con banderas durante un incidente · prioridad de tráfico por recorrido de usuario · experiencia del usuario degradado y su comunicación · probar la degradación antes de necesitarla

LABORATORIO: Clasificar las funciones del caso por criticidad e implementar con banderas de OpenFeature una degradación que mantenga la reserva de turnos con el proveedor de pagos caído, midiendo el SLO del recorrido crítico durante la falla

MÓDULO 4 CARGA Y CAPACIDAD

6 h · 2 sesiones

11 Pruebas de carga bien modeladas

modelo abierto frente a modelo cerrado · omisión coordinada y por qué miente el percentil · perfil de carga a partir del tráfico real · criterios de aprobación expresados en SLO · carga sostenida, pico y prueba de remojo · el generador de carga como cuello de botella · k6: ejecutores de tasa de llegada constante

LABORATORIO: Escribir con k6 una prueba de modelo abierto que reproduzca el perfil del caso, demostrar con la misma carga que el modelo cerrado ocultaba la degradación, y fijar el punto donde el servicio deja de cumplir su SLO

12 Planificación de capacidad contra el SLO

pronóstico de demanda con estacionalidad · margen para perder una zona sin romper el SLO · cuotas y límites del proveedor como techo real · capacidad orgánica frente a capacidad por eventos · costo del margen y su negociación · revisión de capacidad trimestral

LABORATORIO: Producir el plan de capacidad del caso para la próxima temporada alta con pronóstico, margen para perder una zona, cuotas que hay que pedir y costo del margen, validado con una prueba de carga al nivel pronosticado

MÓDULO 5 RECUPERACIÓN Y CAOS

6 h · 2 sesiones

13 Varias zonas, objetivos de recuperación y la prueba de restaurar

dominios de falla: proceso, nodo, zona y región · objetivos de tiempo y de punto de recuperación por recorrido · activo-pasivo y activo-activo con su costo · respaldo que nunca se restauró · simulacro de recuperación con reloj · documentación de emergencia accesible sin la plataforma

LABORATORIO: Ejecutar un simulacro de pérdida de zona y de restauración de datos del caso, midiendo el tiempo y el punto de recuperación reales contra los objetivos declarados; extensión opcional en dos zonas de una nube con presupuesto y alertas, destruida al cierre

14 Ingeniería del caos con hipótesis

estado estable expresado en SLI · hipótesis, radio de impacto y criterio de aborto · fallas de red, de proceso y de recursos con Toxiproxy · empezar fuera de producción y subir de nivel · experimento que no encuentra nada también es resultado · convertir un hallazgo en acción verificable

LABORATORIO: Diseñar y ejecutar con Toxiproxy dos experimentos de caos sobre el caso con hipótesis escrita, radio de impacto y criterio de aborto automático por SLI, y convertir cada hallazgo en una acción con su verificación

MÓDULO 6 CAMBIO SEGURO

6 h · 2 sesiones

15 Entrega progresiva atada al presupuesto

la mayoría de los incidentes empiezan con un cambio · canario con tráfico ponderado en Envoy · análisis del canario comparando SLI · banderas de característica para separar desplegar de publicar · puerta de despliegue según el presupuesto restante · reversión automática y su verificación

LABORATORIO: Publicar una versión defectuosa del caso como canario con tráfico ponderado en Envoy y demostrar que el análisis por SLI la revierte antes de consumir el 2 % del presupuesto, y que con el presupuesto agotado la puerta bloquea el despliegue

16 Revisión de preparación para producción

qué debe tener un servicio antes de que SRE lo opere · lista de verificación: SLO, alertas, manuales, capacidad y recuperación · dependencias declaradas y probadas · riesgo de un cambio y su clasificación · congelamientos por calendario y por presupuesto · devolver un servicio a su equipo de desarrollo

LABORATORIO: Ejecutar la revisión de preparación para producción sobre un servicio nuevo del caso, entregar la lista de brechas con prioridad y decidir, con argumento, si SRE lo acepta

MÓDULO 7 INCIDENTES Y PERSONAS

15 h · 5 sesiones

17 Gestión de incidentes

severidades definidas por impacto en el usuario · roles: responsable del incidente, operaciones y comunicación · canal, línea de tiempo y decisiones registradas · comunicación hacia afuera con cadencia fija · escalar sin culpa y pedir ayuda a tiempo · mitigar primero, entender después · cerrar un incidente: criterio explícito

LABORATORIO: Coordinar un incidente simulado del caso con roles rotativos y una página de estado, registrando la línea de tiempo y cumpliendo la cadencia de comunicación declarada, y evaluar la coordinación contra una lista de verificación

18 Postmortem sin culpables

la cultura sin culpables y lo que no significa · línea de tiempo con evidencia, no con memoria · factores contribuyentes en lugar de causa raíz única · acciones con responsable, fecha y verificación · lo que salió bien también se escribe · revisión del postmortem en grupo · seguimiento de acciones y su tasa de cierre

LABORATORIO: Escribir el postmortem del incidente de la sesión 17 con línea de tiempo respaldada por telemetría, factores contribuyentes y acciones verificables, y revisarlo en grupo con un formulario de calidad

19 Guardias sostenibles

carga de guardia medida: avisos por turno y horas fuera de horario · rotaciones, relevo y seguimiento del sol · manuales de respuesta por síntoma · compensación y límites saludables · avisos que se eliminan, se automatizan o se degradan a ticket · incorporar a alguien nuevo a la guardia

LABORATORIO: Analizar un mes de avisos simulados del caso, eliminar o degradar los que no son accionables, escribir dos manuales por síntoma y demostrar que la carga de guardia baja al menos a la mitad

20 Trabajo repetitivo y automatización

qué es toil y qué no lo es · medir el trabajo repetitivo del equipo · priorizar la automatización por tiempo recuperado · automatización segura: idempotencia, límites y registro · automatizar el diagnóstico antes que la acción · el tope de toil y su defensa ante la gerencia

LABORATORIO: Medir el trabajo repetitivo de una semana de operación simulada del caso, automatizar la tarea de mayor retorno con límites y registro, y calcular las horas mensuales recuperadas

21 **La confiabilidad como práctica de la organización**

revisiones periódicas de SLO con producto · informes de confiabilidad que leen los directivos · SRE embebido, centralizado o consultivo · introducir SLO en una organización que no los tiene · métricas del equipo: tiempo de detección, de mitigación y tasa de acciones cerradas · confiabilidad de sistemas con modelos de IA: qué cambia en el SLI

LABORATORIO: Preparar la revisión trimestral de confiabilidad del caso para producto y dirección, con SLO, presupuesto consumido, incidentes, acciones y una decisión que se pide al negocio

---

MÓDULO 8 **PROYECTO INTEGRADOR** 9 h · 3 sesiones

22 **Game day sin aviso**

diseño del ejercicio por un equipo que no responde · tres fallas inyectadas sin aviso previo · reglas de seguridad y criterio de corte del ejercicio · respuesta con roles y comunicación · medición de detección, mitigación y presupuesto consumido · registro para el postmortem

LABORATORIO: Responder a un game day con tres fallas inyectadas sin aviso por otro equipo, detectándolas por alertas de SLO, coordinando con roles y registrando tiempos de detección y mitigación y presupuesto consumido

23 **Postmortem del game day y verificación de acciones**

postmortem de tres fallas encadenadas · acciones priorizadas por presupuesto ahorrado · implementación de las acciones de mayor retorno · verificación: repetir la falla y medir · actualización de SLO, alertas y manuales · comunicar el resultado a los equipos y a producto

LABORATORIO: Escribir el postmortem del game day, implementar las dos acciones de mayor retorno y verificarlas repitiendo las fallas y midiendo el presupuesto consumido antes y después

24 **Defensa del servicio**

SLO elegidos y su negociación · alertas, tableros y tiempo de diagnóstico · patrones y degradación medidos · plan de capacidad y simulacro de recuperación · postmortem y acciones verificadas · preguntas cruzadas del grupo

LABORATORIO: Defender ante el grupo el servicio del caso con sus SLO, alertas, resultados del game day y acciones verificadas, respondiendo las preguntas cruzadas con datos del propio laboratorio

---

LO QUE VAS A USAR

Herramientas libres que son medios del método, no el temario.

Servicio y entorno

- Contenedores con Docker Compose
- Servicio de ejemplo de varios componentes
- Envoy como proxy

SLO

- OpenSLO
- Sloth para generar reglas de alerta

Telemetría

- OpenTelemetry y su colector
- Prometheus y Alertmanager
- Jaeger
- Loki
- Grafana

Carga y fallas

- k6
- Toxiproxy

Cambio y degradación

- Tráfico ponderado en Envoy
- OpenFeature con flagd

Práctica de incidentes

- Página de estado del caso
- Plantilla de postmortem
- Manuales por síntoma

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 servicio que se rompe en cada sesión y decisiones que se defienden con datos.

01 **Clases en vivo por Zoom** 24 sesiones de 3 horas, 2 veces por semana. Se opera el servicio en vivo y se discuten las decisiones 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 **35% concepto / 65% práctica** Cada concepto se aplica en el mismo día sobre el servicio del caso, con una falla inyectada para verificarlo.

04 **Antes y después, bajo la misma falla** Ningún patrón se da por bueno porque se aplicó: se mide el presupuesto de error consumido con y sin él.

05 **Incidentes reales del laboratorio** Los postmortems se escriben sobre lo que pasó en tu entorno, no sobre casos inventados, y sus acciones se verifican.

06 **Game day sin aviso** Otro equipo diseña tres fallas que no conoces. Las respuestas se miden en tiempo de detección, de mitigación y presupuesto consumido.

Horarios · Primer grupo

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

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

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

REQUISITOS

Lo que necesitas traer a la sesión 01.

No hay nivelación previa. La sesión 01 levanta el servicio, lo rompe y pregunta qué vio cada uno.

Operación

- ✓ Haber operado o desplegado un servicio web
- ✓ Haber vivido al menos un incidente, aunque sea de lejos
- ✓ Nociones de una nube pública

Web y sistemas

- ✓ HTTP, latencia y códigos de error
- ✓ Linux en terminal
- ✓ Contenedores a nivel de uso

Programación

- ✓ Un lenguaje para leer y modificar un servicio: Python, Go o Java
- ✓ Leer y escribir YAML
- ✓ Git

Números

- ✓ Porcentajes y percentiles
- ✓ Leer un gráfico de series de tiempo

Equipo y tiempo

- ✓ Computadora con 16 GB de memoria y 4 núcleos
- ✓ Seis horas de clase por semana
- ✓ Entre cuatro y seis horas de práctica
- ✓ 12 semanas de compromiso

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

Estos costos NO están incluidos en el precio del programa. Todo el software del laboratorio es gratuito.

- El laboratorio corre en tu máquina con contenedores y software de código abierto: con un equipo de 16 GB de memoria el costo de nube es cero
- Si tu equipo no alcanza, una máquina virtual de 4 núcleos y 16 GB alquilada solo durante las prácticas cuesta del orden de US$20 en todo el programa
- Una extensión opcional en la sesión 13 despliega el servicio en dos zonas de una nube pública: unos pocos dólares, con presupuesto y alertas configurados en esa sesión y todo destruido al cierre

PRECIO

72 horas para dejar de operar por intuición.

US$400

Pago único · USD · Incluye impuestos

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

Incluye

- ✓ 24 sesiones en vivo por Zoom (72 horas lectivas)
- ✓ Grabación de todas las clases con acceso por 12 meses
- ✓ Repositorio del servicio del caso con su entorno de telemetría listo para levantar
- ✓ Banco de fallas inyectables y escenarios de carga
- ✓ Plantillas de SLO, política de presupuesto, postmortem y revisión de preparación para producción
- ✓ Diseño y conducción del game day sin aviso
- ✓ Los 3 certificados progresivos
- ✓ Canal privado del grupo para dudas entre sesiones
- ✓ Actualizaciones del material por 12 meses

No incluye

- — El equipo de trabajo: necesitas una computadora con 16 GB de memoria; si no la tienes, alquilar una máquina virtual entra en el rango de costos propios (US$0 a 30)
- — Exámenes ni certificaciones de terceros: este programa no prepara para ninguno
- — Herramientas comerciales de monitoreo: todo el laboratorio usa software libre

Métodos de pago

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

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

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

PREGUNTAS FRECUENTES

Preguntas frecuentes sobre el programa.

¿Las clases son en vivo o grabadas?

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

¿Cuándo empiezan las clases?

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

¿Qué nivel necesito?

Haber operado o desplegado un servicio web, entender HTTP y latencia, y poder leer y modificar un servicio en Python, Go o Java. No hace falta ser SRE: hace falta haber estado cerca de un servicio que falla.

¿Necesito una cuenta de nube?

No. Todo el laboratorio corre en tu máquina con contenedores y software libre. Hay una única extensión opcional en la sesión 13 para probar la pérdida de una zona en una nube pública, que cuesta unos pocos dólares si la haces y se destruye al terminar.

¿Qué equipo necesito?

Una computadora con 16 GB de memoria y 4 núcleos, con Docker. El servicio del caso, su telemetría y el generador de carga corren a la vez. Si tu equipo no alcanza, el rango de costos propios del programa es de US$0 a 30: cero si tu equipo alcanza y del orden de 20 dólares si alquilas una máquina virtual.

Ya cursé un programa de sistemas distribuidos. ¿Me sirve este?

Sí, y conviene que sepas qué se repite. Cerca de un tercio de las sesiones toca problemas que un programa de sistemas distribuidos o de operación en la nube también trata: tiempos de espera y cortacircuitos, inyección de fallas, capacidad, recuperación, canarios, guardias y gestión de incidentes. Aquí el centro es otro: operar un servicio contra un SLO y coordinar a las personas durante el incidente, con otros laboratorios. Lo nuevo serían SLI, SLO, presupuesto de error, alertas por tasa de consumo, postmortems y trabajo repetitivo. No hay descuento ni convalidación entre programas.

¿Qué herramientas se usan y por qué esas?

OpenTelemetry, Prometheus, Grafana, Jaeger, Loki, k6, Toxiproxy, Envoy, Sloth y OpenFeature. Son libres y gratuitas, y alcanzan para todo el método. Grafana, Loki y k6 tienen licencia AGPL, que permite usarlos sin costo en tu propio entorno. Lo que se aprende se traslada a cualquier herramienta comercial.

¿El game day es realmente sin aviso?

Sí. Las tres fallas las diseña otro equipo o el instructor y no se anuncian. Hay reglas de seguridad y un criterio para cortar el ejercicio, pero el objetivo es que la detección dependa de tus alertas y no de saber qué va a pasar.

¿Sirve si mi empresa no tiene un equipo de SRE?

Sí, y es el caso más común. Una sesión entera trata cómo introducir SLO en una organización que no los usa, y el método se aplica aunque la guardia la haga el mismo equipo que desarrolla.

¿Qué pasa si falto a una clase?

La grabación queda disponible y cada laboratorio tiene su punto de partida en el repositorio. Las sesiones de incidentes y el game day se viven en grupo: si faltas a esas, podrás hacer el ejercicio pero no la coordinación con otros.

¿Los certificados tienen validez oficial?

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

¿Cuánto tiempo debo dedicarle por semana?

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

¿Qué métodos de pago aceptan?

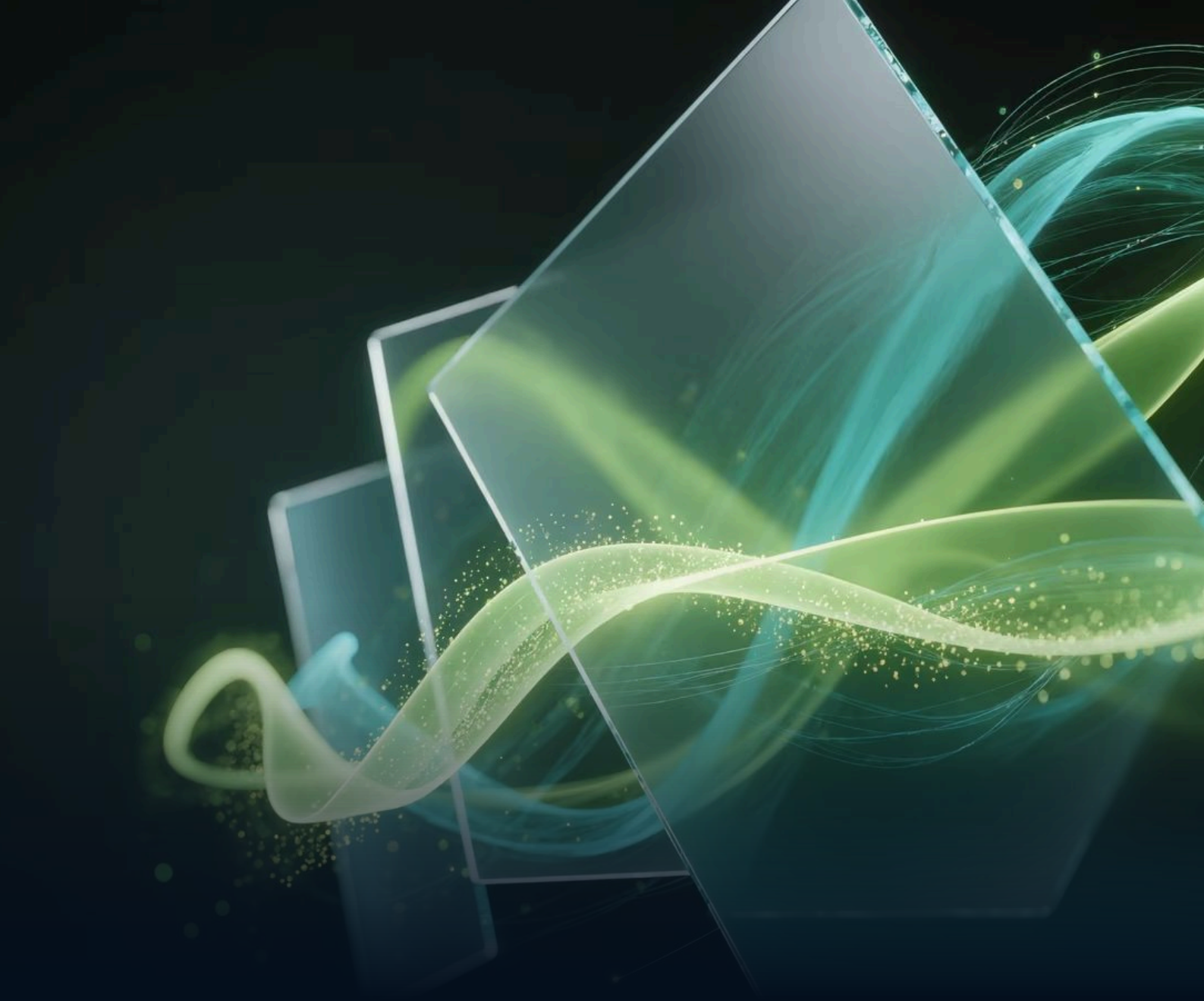
El pago lo procesa Paddle y acepta Tarjeta de crédito, Tarjeta de débito internacional, PayPal, Apple Pay, Google Pay. El cobro es en dólares. Importante: no procesamos Yape, Plin, OXXO, Mercado Pago ni transferencias locales. Si tu banco rechaza compras internacionales, habilítalas antes de pagar o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

¿Emiten factura o comprobante?

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

¿Cuál es la política de reembolso?

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



Reactor
COURSES

Ingeniería de software en la era de los agentes

¿Te quedó alguna duda?

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

`reactorcourses.com/programas/sre-resilience-architecture`

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

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