

Multi-Cloud Engineering Program

Parte 22 — Especializaciones, certificaciones y práctica profesional

12 clases · 52 horas

Cuaderno de una sola parte: su introducción, sus clases completas y sus evaluaciones.

Extracto del manual integral, generado desde las mismas fuentes versionadas.

Alcance de este cuaderno

Componente	Cobertura
Parte	22 de 23
Clases	265–276
Evaluaciones	12
Horas estimadas	52

Índice

Alcance de este cuaderno	2
Índice	3
Parte 22 — Especializaciones, certificaciones y práctica profesional	4
265 — Ruta Cloud Engineer y mapa de competencias	6
266 — Ruta DevOps y Delivery Engineer	16
267 — Ruta Platform Engineer	26
268 — Ruta Site Reliability Engineer	36
269 — Ruta Cloud Security Engineer	47
270 — Ruta FinOps Practitioner	57
271 — Ruta Cloud Data y AI Engineer	68
272 — Ruta Cloud Solutions Architect	79
273 — Mapeo AWS, Azure, Google Cloud, Kubernetes y FinOps	89
274 — Preguntas de escenario y estrategia de examen	99
275 — Portafolio, evidencia, README y entrevista de sistemas	110
276 — Proyecto: defensa técnica ante panel	121

Parte 22 — Especializaciones, certificaciones y práctica profesional

← Parte 21 · Índice completo · Parte 23 →

Descargar: Esta parte en PDF · Manual integral

Nivel: intermedio-avanzado · Clases: 12 · Duración sugerida: 6–8 semanas

Transformar el conocimiento transversal en perfiles demostrables, preparación certificable y comunicación profesional.

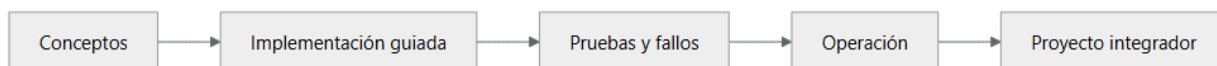
Resultados de la parte

Al completar esta parte podrás explicar el dominio con lenguaje independiente del proveedor, implementar sus mecanismos esenciales, diagnosticar fallos y defender decisiones mediante evidencia, costo, seguridad y objetivos de confiabilidad.

Modelo mental

Una especialización combina fundamentos, evidencia de proyectos y juicio bajo restricciones; una insignia sin práctica no sustituye esa combinación.

Secuencia



Clases

ID	Clase	Laboratorio	Horas
265	Ruta Cloud Engineer y mapa de competencias	decision	4
266	Ruta DevOps y Delivery Engineer	decision	4
267	Ruta Platform Engineer	decision	4
268	Ruta Site Reliability Engineer	decision	4
269	Ruta Cloud Security Engineer	decision	4
270	Ruta FinOps Practitioner	decision	4
271	Ruta Cloud Data y AI Engineer	decision	4
272	Ruta Cloud Solutions Architect	decision	4
273	Mapeo AWS, Azure, Google Cloud, Kubernetes y FinOps	assessment	4
274	Preguntas de escenario y estrategia de examen	assessment	4
275	Portafolio, evidencia, README y entrevista de sistemas	assessment	4
276	Proyecto: defensa técnica ante panel	capstone	8

Evaluación de la parte

- 35 % laboratorios y evidencia reproducible.
- 25 % retos verificables.
- 20 % decisiones y ADR.
- 10 % seguridad, confiabilidad y costo.

- 10 % proyecto integrador de la parte.

Se aprueba con 80 % de clases en nivel B o superior y el proyecto integrador reproducible.

Pauta bibliográfica

- The Staff Engineer's Path — Tanya Reilly.
- The Manager's Path — Camille Fournier.
- Staff Engineer — Will Larson.

265 — Ruta Cloud Engineer y mapa de competencias

Parte: 22 — Especializaciones, certificaciones y práctica profesional Nivel: intermedio-avanzado · Horas estimadas: 4
Laboratorio: decision · Estado: EXECUTABLECORE

Propósito

Dar el mapa de competencias que ordena la parte 22 y la ruta base de la que salen todas las demás. La clase separa lo que hay que saber de lo que hay que saber hacer, distingue las competencias atadas a un producto de las atadas a una restricción —que son las que duran—, y da el método para evaluarse sin engañarse: por lo que uno ha resuelto, no por lo que ha leído.

Resultados de aprendizaje

Al finalizar podrás:

1. Distinguir competencias de producto y competencias de restricción.
2. Situar las ocho rutas de la parte 22 en un mapa común.
3. Evaluar el nivel propio por evidencia y no por familiaridad.
4. Construir un plan de aprendizaje con hueco identificado y prueba.
5. Reconocer el techo de cada ruta y qué lo levanta.

Conceptos centrales

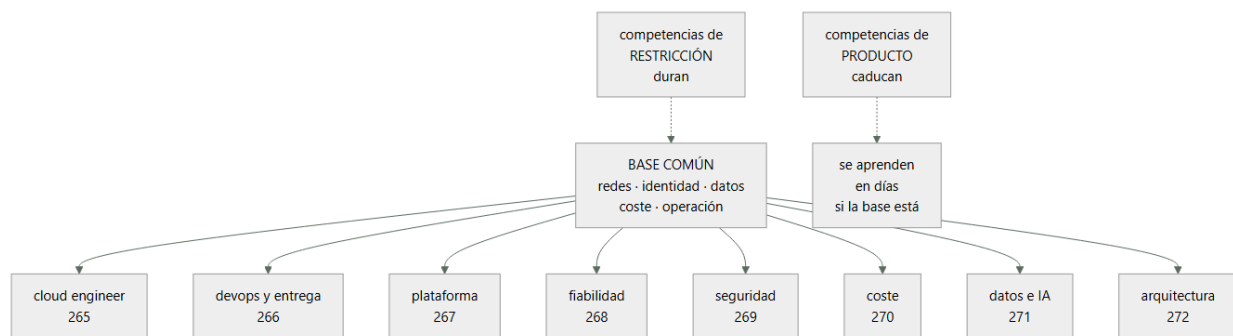
Concepto	Comprensión verificable
competencia de restricción	Saber que se organiza alrededor de un límite del mundo: latencia, coherencia, coste, superficie de ataque. Sobrevive a los productos.
competencia de producto	Saber cómo se hace algo en un servicio concreto. Caduca con él.
nivel por evidencia	Lo que uno ha resuelto sin ayuda, con las consecuencias delante. La única medida honesta.
ilusión de familiaridad	Confundir haber leído o visto algo con saber hacerlo.
ruta	Combinación de competencias que un puesto exige. No es una identidad: es un reparto de énfasis.
techo de ruta	El límite al que se llega sin adquirir competencias de otra ruta.

Modelo mental

Una especialización combina fundamentos, evidencia de proyectos y juicio bajo restricciones; una insignia sin práctica no sustituye esa combinación.

Aplicado a esta clase, separa siempre cuatro planos: intención (qué necesita el usuario), configuración (qué declaramos), estado observado (qué existe de verdad) y evidencia (cómo sabemos que cumple). Confundirlos produce diseños que se ven correctos en un diagrama pero fallan al operar.

■ Flujo de razonamiento



Desarrollo

1. Restricción frente a producto

La distinción que ordena una carrera entera: hay saberes que caducan con el servicio que los origina y saberes que no.

COMPETENCIA DE PRODUCTO

cómo se configura un balanceador en la nube X
qué nombre tiene el servicio de colas en la nube Y
qué campos lleva una política en la nube Z

- se aprende en días si la base está
- caduca cuando el producto cambia
- y es lo que más ocupa en los temarios

COMPETENCIA DE RESTRICCIÓN

por qué la latencia entre regiones limita la coherencia
clases 183, 187
por qué un reintento sin retroceso multiplica la carga
clase 201
por qué el coste se decide al diseñar
ley 14
por qué un permiso amplio es una decisión de riesgo
clase 231
por qué una cola cambia dónde está la garantía
ley 18

→ no caduca, porque la restricción no depende del producto
→ un producto nuevo es otra forma de encajar el mismo límite

Y la consecuencia práctica, que es lo que cambia decisiones:

QUIEN SABE LA RESTRICCIÓN APRENDE EL PRODUCTO EN DÍAS
QUIEN SABE EL PRODUCTO NO DEDUCE LA RESTRICCIÓN NUNCA

- y por eso este programa ha enseñado por restricción y ha usado los tres proveedores como ejemplos
- y por eso la parte 19 pudo comparar las tres nubes midiendo, en vez de listando

Y las cinco restricciones que estructuran todo lo demás:

- 1 LA DISTANCIA
la luz tarda; y de ahí salen latencia, coherencia, regiones, réplicas y conmutación
- 2 EL FALLO PARCIAL
las cosas fallan a medias y lento, no del todo
→ y de ahí, plazos, reintentos, cortacircuitos y degradación
clases 185, 201
- 3 LA IDENTIDAD
todo acceso es una afirmación que hay que verificar
→ y de ahí, autenticación, autorización, privilegio mínimo y auditoría
clase 231
- 4 EL COSTE POR USO
cada operación se factura
→ y de ahí, arquitectura, formatos, y la ley 28

LA PRUEBA QUE LO DESHACE
«¿cuándo lo hiciste sin ayuda?»

«¿qué salió mal y cómo lo supiste?»

→ y esto duele, y es útil

por cada competencia

sí / no

¿en producción o en un ejercicio?

¿qué mide que salió bien?

→ los ejercicios que no fallan no enseñan

1 ELIGE UNA RESTRICCIÓN, no un producto

«no sé razonar sobre coherencia entre regiones»

en tu sistema, en un proyecto propio o en un ejercicio

qué crees que va a pasar

y compara con lo que creías

→ y ahí tienes evidencia para el portafolio

→ y este ciclo, repetido, es lo que produce el nivel 3

→ leer más contenidos no lo produce nunca

La ruta base: construir y sostener servicios en la nube

aprovisiona y mantiene infraestructura

monta redes, cómputo, almacenamiento y bases

automatiza con infraestructura como código clase 128

responde de que funcione

```
nivel 1 ejecuta cambios definidos por otros
```

nivel 2	diseña y monta un servicio completo, con red,
---------	---

identidad, copias y señales

nivel 3 decide entre alternativas con coste y riesgo

medidos, y lo defiende

nivel 4 define cómo se construye en la organización

redes: por qué no llega el tráfico

identidad y permisos clase 231

infraestructura como código, con estado y módulos

Infraestrutura como serviço, sem estado, modulares
classes 128, 232

copias y restauración probada

coste al diseñar	clase 220
------------------	-----------

clase 218	clase 218
y diagnóstico	clase 258

• O teste, que se é importante de toda classe.

EL TECHO DE LA RUTA BASE

se llega a construir bien lo que a uno le piden
→ y ahí se para

y lo que lo levanta NO es más profundidad técnica
es una de estas tres

- a saber decidir entre opciones y defenderlo
→ ruta de arquitectura clase 272
- b hacer que otros vayan más rápido
→ ruta de plataforma clase 267
- c responder de una propiedad del sistema entero
→ fiabilidad, seguridad o coste
clases 268, 269, 270

→ y ninguna de las tres es «saber más servicios»
→ el hueco entre nivel 2 y 3 es de JUICIO, no de catálogo
→ que es exactamente lo que la ley 30 dijo de los
procedimientos: lo que falta no es el paso, es la
decisión clase 264

Y el consejo que más cambia trayectorias:

ESPECIALIZARSE EN UNA NUBE A FONDO Y APRENDER A TRADUCIR

→ mejor que las tres por encima
→ porque la profundidad en una enseña la restricción,
y la restricción se traduce clase 273
→ y quien sabe las tres por encima no ha visto ninguna
fallar de verdad

y la trampa del currículo
listar veinte servicios no dice nada
→ lo que dice algo es «reduje el tiempo de recuperación
de 47 a 11 minutos y así lo medí» clase 275

Y la lista de comprobación de la clase:

- sé distinguir qué de lo que sé es producto y qué es restricción
- tengo la base común, no solo la parte que uso
- mi nivel lo he fijado por evidencia, no por familiaridad
- por cada competencia puedo decir qué salió mal alguna vez
- mi plan parte de una restricción, no de un producto
- tengo un problema real donde practicarla
- escribo la hipótesis antes y la comparo después
- sé cuál es el techo de mi ruta y qué lo levanta
- tengo una nube a fondo antes que tres por encima
- mis logros están escritos con cifras, no con servicios

Y el cierre que enlaza con la clase siguiente: la primera especialización que sale de la base es la que se ocupa de que el cambio llegue a producción rápido y sin romper. Entrega continua y su ruta es la materia de la clase 266.

Ejemplo trabajado

Tres autoevaluaciones reales del equipo de CloudShop, hechas con la rejilla por evidencia. Lo que sigue es lo que la rejilla dijo frente a lo que cada persona creía, y el plan que salió de ahí.

Persona A · Cuatro años de experiencia. Se creía nivel 3.

autoevaluación inicial, por familiaridad

redes	«alto»
identidad	«alto»
infraestructura como código	«alto»
copias	«medio»
coste	«medio»
diagnóstico	«alto»

y la rejilla por evidencia

	¿solo?	¿producción?	¿qué falló?
redes	sí	sí	nada
identidad	sí	sí	nada
infraestructura como código	sí	sí	«un estado corrupto»
copias	no	-	-

coste	no	-	-
diagnóstico	sí	sí	«tardé 3 h en un caso»

Y la conversación que lo cambió:

«¿qué falló en redes?»	«nada»
«¿has visto una asimetría de rutas?»	no
«¿has diagnosticado un problema de unidad máxima de transmisión?»	no
«¿has visto un nombre resolver distinto según desde dónde se pregunte?»	no

→ nivel real en redes: 2, no 3
→ había construido muchas redes y no había roto ninguna
→ y el nivel 3 exige haber visto fallar

y en copias, el hallazgo
«las copias las configuré yo»
«¿has restaurado alguna?» no
→ nivel 1, no medio ley 22

Y el plan que salió:

restricción elegida	el fallo parcial
problema real	los ensayos de la clase 261
hipótesis escrita	«al añadir 200 ms, el flujo se degradará proporcionalmente»
resultado	4.100 ms; hipótesis refutada

→ y esa refutación produjo más aprendizaje que dos años construyendo redes que funcionaban
→ nivel en redes y en fallo parcial, 12 meses después: 3

Persona B · Once meses. Se creía nivel 1 en todo.

la rejilla dio otra cosa

	¿solo?	¿producción?	¿qué falló?
diagnóstico	sí	sí	«seguí una hipótesis falsa 2 h»
procedimientos	sí	sí	«el mío estaba mal escrito y otro no lo entendió»
copias	sí	sí	«restauré y faltaban 40 min»

→ nivel 2 en tres competencias
→ y nivel 1 en redes e identidad, correctamente
→ había hecho MENOS cosas y las había hecho fallar
→ que es lo que produce nivel

Y lo que esto reveló del equipo:

quien lleva menos tiempo suele tener MEJOR calibrada la autoevaluación
→ porque recuerda cuándo pidió ayuda

y quien lleva más tiempo confunde antigüedad con nivel
→ porque ha visto muchas veces lo mismo funcionar

→ la rejilla por evidencia corrige a los dos

Persona C · Ocho años. Nivel 3 real, con techo.

la rejilla confirmó nivel 3 en casi todo	
decide entre alternativas con cifras	sí
anticipa modos de fallo	sí
defiende decisiones ante negocio	sí

y la pregunta del techo
«¿cuál fue tu último trabajo que cambió cómo trabaja

otro equipo?»
→ ninguno en dos años

→ nivel 3 estable, techo alcanzado
→ y su queja era «no avanzo, y ya sé más que nadie de esto»
→ que es el síntoma exacto del techo de la ruta base

Y las tres salidas que se le plantearon, con lo que cada una exigía:

- a ARQUITECTURA clase 272
lo que le falta: defender ante un panel que no es técnico, y escribir decisiones para que sobrevivan a su autor
- b PLATAFORMA clase 267
lo que le falta: tratar a otros equipos como clientes y medir adopción, no calidad interna
- c FIABILIDAD clase 268
lo que le falta: responder de un número acordado con negocio, y decir que no a un despliegue

eligió (b)
y a los 9 meses, su medida era otra
equipos que despliegan solos 3 → 14
tiempo de un servicio nuevo 3 semanas → 2 días

→ el mismo conocimiento técnico, otro efecto
→ y eso es lo que distingue el nivel 4

Y la comparación agregada del equipo, 9 personas.

redes	media 2,9	media 2,1
identidad	2,7	1,9
infraestructura como código	3,1	2,6
copias y restauración	2,4	1,2
coste	2,2	1,1
diagnóstico	2,8	2,3
operación	2,6	1,8

→ el hueco es mayor donde la práctica es más rara
→ copias y coste son los peores, y no es casualidad:
son los que solo se practican cuando algo va mal
→ y la parte 21 lo confirmó: la restauración real tardó
3 h 10 frente a los 40 minutos que todos creían

Y lo que el equipo hizo con esa tabla:

no se usó para evaluar a nadie
→ se usó para decidir qué ensayar
→ y los ensayos de la clase 261 se ordenaron por el hueco más grande

12 meses después
copias y restauración 1,2 → 2,4
coste 1,1 → 2,0
y el tiempo de restauración medido 3 h 10 → 52 min

→ la rejilla no midió el aprendizaje: lo dirigió

La lección que esta clase deja: la persona con cuatro años se creía nivel 3 en redes porque nunca había visto una fallar, y la de once meses tenía nivel 2 real en tres competencias porque las había roto. Y el hueco entre autoevaluación y evidencia era mayor justo en copias y coste, que son las dos competencias que solo se practican cuando algo va mal.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-22-specializations-certifications-career/265-ruta-cloud-engineer-y-mapa-de-competencias/lab.py
```

El laboratorio selecciona el motor de práctica decision y produce labresult.json. El escenario, sus comprobaciones y el artefacto esperado corresponden a esta clase; no requiere credenciales y deja explícito qué debe revalidarse en un

sandbox real.

1. Lee exercise.steps y formula una predicción antes de ejecutar.
2. Ejecuta la práctica y verifica todos los elementos de checks.
3. Provoca el caso de negativetest y explica la señal observada.
4. Materializa cloud-engineer-plan en evidence/ usando la plantilla indicada.
5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

El artefacto principal es una matriz de decisión ponderada y un ADR. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye cloud-engineer-plan para el caso CloudShop. Incluye una alternativa descartada, un supuesto que pueda falsarse, una prueba de fallo y una decisión de rollback.

■ Criterio de aceptación

- [] lab.py termina con código 0 y genera JSON válido.
- [] La entrega conecta al menos tres requisitos con mecanismos verificables.
- [] Existe una prueba positiva y una prueba negativa con evidencia.
- [] Seguridad, costo y operación aparecen como decisiones, no como anexos.
- [] Se declara una limitación y una condición que obligaría a revisar el diseño.
- [] Otra persona puede repetir el recorrido sin conocimiento tácito.

■ Errores frecuentes

Síntoma	Causa probable	Corrección
Se estudia mucho y el nivel no sube	Leer produce reconocimiento, no capacidad; la familiaridad se confunde con saber	Por cada competencia responde qué hiciste solo, qué salió mal y qué harías distinto; si no hay respuesta, el nivel es 1.
Se conocen tres nubes y no se resuelve bien en ninguna	La amplitud sin profundidad no enseña la restricción, que es lo que se traduce	Domina una nube hasta haberla visto fallar y luego traduce; la restricción es transferible, la sintaxis se aprende en días.
Se llega a construir bien y la carrera se estanca	Es el techo de la ruta base; no se levanta con más servicios sino con juicio o alcance	Elige entre decidir y defender, hacer que otros vayan más rápido, o responder de una propiedad del sistema entero.
El currículo lista veinte servicios y no consigue entrevistas técnicas buenas	Una lista de productos no demuestra capacidad de resolver	Escribe logros con cifra y método: qué medías antes, qué cambiaste y qué mide que funcionó.
La autoevaluación del equipo no coincide con lo que ocurre en incidentes	Se evaluó por antigüedad y familiaridad, no por evidencia	Usa la rejilla por evidencia y ordena los ensayos por el hueco mayor; suele estar en copias y coste.
Un ejercicio de práctica salió perfecto y no se aprendió nada	El ejercicio era demasiado pequeño para fallar	Escoge problemas con consecuencias medibles y escribe la hipótesis antes; lo que no falla no enseña.

■ Seguridad, ética y costo

Trabaja con cuentas propias o sandboxes autorizados. No publiques secretos, identificadores reales ni datos personales. Antes de crear recursos pagos define presupuesto, etiquetas y comando de destrucción; después verifica que no queden recursos huérfanos. Los ejemplos locales enseñan contratos, pero no certifican cumplimiento ni disponibilidad de producción.

■ Preguntas de comprobación

1. ¿Qué distingue una competencia de restricción de una de producto?
2. ¿Cuáles son las cinco restricciones que estructuran el resto?
3. ¿Cómo se fija el nivel propio sin caer en la ilusión de familiaridad?
4. ¿Qué separa el nivel 2 del nivel 3 y por qué es el salto que menos gente da?
5. ¿Cuál es el techo de la ruta base y qué tres cosas lo levantan?

Referencias

- Ericsson, K. A. y Pool, R. (2016). Peak: secrets from the new science of expertise. <<https://www.hachettebookgroup.com/titles/anders-ericsson/peak/9780544456259/>>
- Dreyfus, S. (2004). The five-stage model of adult skill acquisition. <<https://journals.sagepub.com/doi/10.1177/0270467604264992>>
- AWS (2024). Skills and certification paths. <<https://aws.amazon.com/training/learn-about/>>
- Microsoft (2024). Azure career and skills paths. <<https://learn.microsoft.com/credentials/browse/>>
- Google Cloud (2024). Cloud learning paths by role. <<https://cloud.google.com/learn/training>>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 22 en PDF · Manual integral

Anterior	Índice	Siguiente
← 264 · Proyecto: centro de operaciones de CloudShop	Parte 22 · Programa	266 · Ruta DevOps y Delivery Engineer →

Evaluación — 265 Ruta Cloud Engineer y mapa de competencias

Preguntas

1. Explica el problema que resuelve esta capacidad sin mencionar un proveedor.
2. Identifica una frontera de responsabilidad y su propietario.
3. Compara dos alternativas mediante confiabilidad, seguridad, costo y operación.
4. ¿Qué demuestra el laboratorio y qué no puede demostrar?
5. Propón un caso límite o una prueba de fallo.

Reto verificable

Entrega cloud-engineer-plan con diagrama o configuración, evidencia de ejecución, alternativa descartada, riesgo residual y criterio de rollback.

Criterio de aceptación

- [] Resultado reproducible por una segunda persona.
- [] Evidencia vinculada a cada afirmación técnica.

- [] Prueba negativa o de fallo incluida.
- [] Costos y permisos expresados con unidades y alcance.
- [] Limitaciones declaradas sin presentar la simulación como producción.

Escala

Nivel	Evidencia
A — competente	Cumple todo y defiende trade-offs con datos.
B — en desarrollo	Cumple lo esencial; falta profundidad en una dimensión.
C — inicial	Artefacto parcial o conclusión sin evidencia suficiente.
0	Sin entrega reproducible.

266 — Ruta DevOps y Delivery Engineer

Parte: 22 — Especializaciones, certificaciones y práctica profesional Nivel: intermedio-avanzado · Horas estimadas: 4
Laboratorio: decision · Estado: EXECUTABLECORE

Propósito

La ruta de entrega: hacer que el cambio llegue a producción rápido y sin romper. La clase separa lo que esta especialidad hace de verdad —que no es montar cadenas de construcción— de lo que aparenta, da las competencias que se miden en una entrevista y en el trabajo, y marca su modo de fracaso característico: convertirse en el equipo que aprueba en vez del que habilita.

Resultados de aprendizaje

Al finalizar podrás:

1. Definir qué hace esta ruta y qué no, con las cuatro métricas.
2. Identificar las competencias que se miden y en qué orden crecen.
3. Diagnosticar una cadena de entrega lenta o poco fiable.
4. Evitar el modo de fracaso de convertirse en portero.
5. Reconocer el techo de la ruta y qué lo levanta.

Conceptos centrales

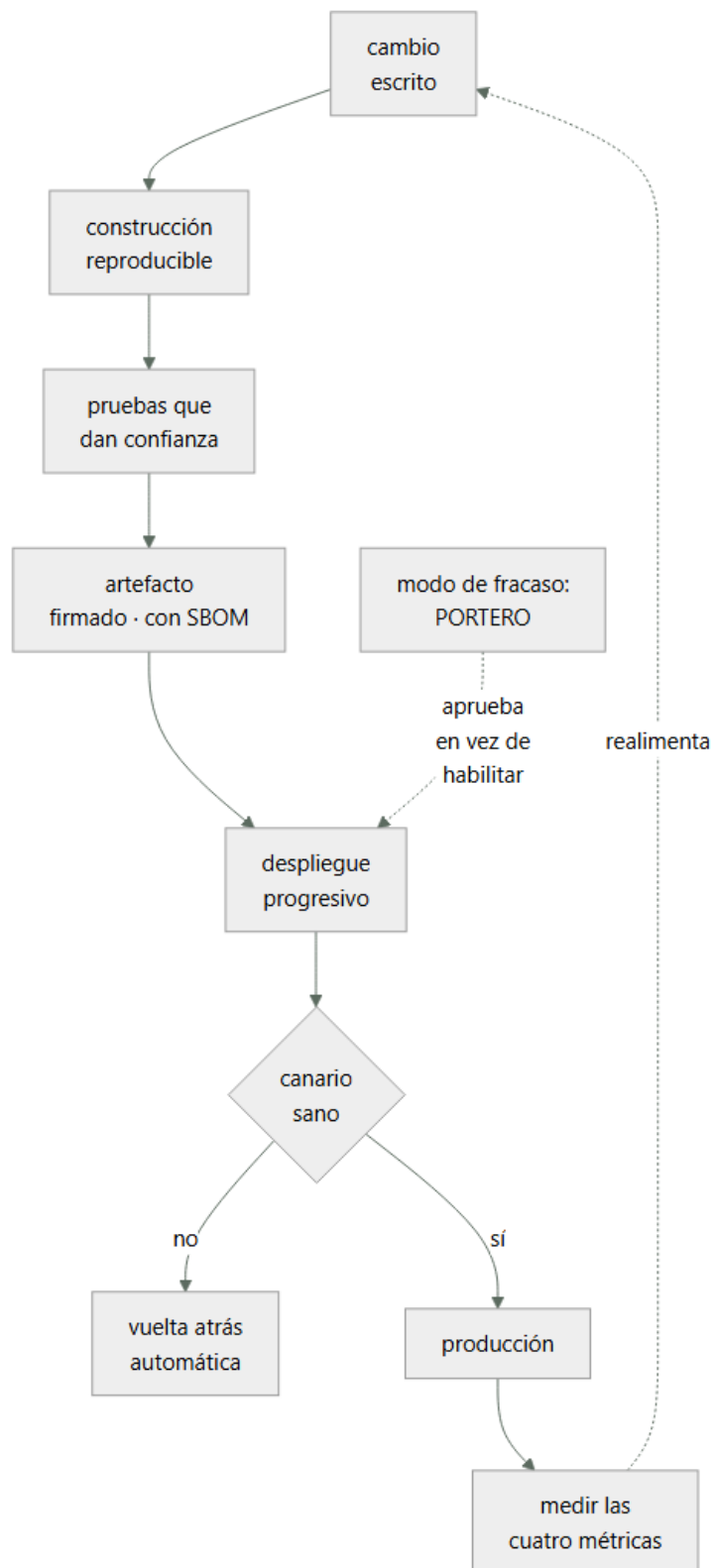
Concepto	Comprensión verificable
ruta de entrega	Especialidad centrada en el camino desde que se escribe un cambio hasta que sirve a un usuario.
cadena de entrega	Construcción, pruebas, empaquetado, despliegue y verificación. El producto de esta ruta.
portero	Modo de fracaso en que el equipo pasa de habilitar a autorizar, y se convierte en el cuello de botella.
tiempo de entrega	De confirmar un cambio a que sirva tráfico. La métrica que más revela.
camino pavimentado	La forma fácil de hacer lo correcto. Reemplaza a la norma que hay que recordar.
prueba que da confianza	La que, si pasa, permite desplegar sin mirar. Cualquier otra es decoración.

Modelo mental

Una especialización combina fundamentos, evidencia de proyectos y juicio bajo restricciones; una insignia sin práctica no sustituye esa combinación.

Aplicado a esta clase, separa siempre cuatro planos: intención (qué necesita el usuario), configuración (qué declaramos), estado observado (qué existe de verdad) y evidencia (cómo sabemos que cumple). Confundirlos produce diseños que se ven correctos en un diagrama pero fallan al operar.

■ Flujo de razonamiento



Desarrollo

1. Qué hace esta ruta de verdad

El nombre y las herramientas engañan. Esta ruta no se define por montar cadenas: se define por un número.

EL TRABAJO ES REDUCIR EL TIEMPO Y EL RIESGO ENTRE
«ESTÁ ESCRITO» Y «SIRVE A UN USUARIO»

y se mide con cuatro cosas

clase 260

frecuencia de despliegue
tiempo de entrega
tasa de fallo por cambio
tiempo de recuperación

→ si estas cuatro no mejoran, el trabajo no está produciendo efecto, por muchas herramientas que haya

Y lo que hace, desglosado:

CONSTRUCCIÓN REPRODUCIBLE

la misma entrada produce la misma salida
dependencias fijadas y almacenadas clase 216
y construcción sin acceso a producción

PRUEBAS QUE DAN CONFIANZA

la pregunta no es «¿cuánta cobertura?»
es «si pasan, ¿despliego sin mirar?»
→ y si la respuesta es no, sobra o falta algo

ARTEFACTO ÚNICO Y TRAZABLE

se construye una vez y se promociona
→ no se reconstruye por entorno
firmado, con inventario de componentes clase 216

DESPLIEGUE PROGRESIVO

canario, porcentajes, vuelta atrás automática clase 102
y despliegue separado de activación clase 105

VERIFICACIÓN AUTOMÁTICA

el propio despliegue comprueba que funcionó
→ y si no, revierte solo

Y EL CAMINO PAVIMENTADO

que hacer lo correcto sea lo más fácil
→ una plantilla que ya trae señales, copias, permisos
mínimos y despliegue progresivo
→ esto es lo que hace que la norma no haga falta
ley 16

Y lo que NO es esta ruta, aunque se lo asignen:

no es «el equipo que despliega»
→ si esta ruta despliega por los demás, es el cuello de botella
no es «el equipo de las herramientas»
→ las herramientas son el medio
no es «el que arregla la cadena cuando se rompe»
→ eso es síntoma de una cadena frágil
y no es el que aprueba
→ ese es el modo de fracaso

2. Las competencias que se miden

Lo que distingue a alguien bueno en esta ruta, por orden de aparición.

NIVEL 2 · RESUELVO

monta una cadena completa para un servicio
diagnostica por qué una construcción no es reproducible
distingue prueba útil de prueba decorativa
gestiona secretos sin ponerlos en el repositorio clase 197
y sabe qué hace lento el ciclo, midiéndolo

NIVEL 3 · DISEÑO

decide la estrategia de ramas y de versiones y la defiende con datos de tiempo de entrega
diseña el despliegue progresivo por servicio, según su riesgo clase 260
hace reversible lo que no lo parece: esquemas, mensajes, contratos clases 106, 209, 210
reduce el lote sin bajar la calidad
y monta la cadena de suministro: firma, procedencia, inventario clase 216

NIVEL 4 · CAMBIO EL SISTEMA

convierte el camino pavimentado en el más fácil, y la adopción se mide
elimina el proceso de aprobación demostrando que no filtra riesgo clase 260
y consigue que los equipos de producto respondan de sus despliegues

Y las preguntas que separan niveles en una entrevista:

«¿cuál es vuestro tiempo de entrega y cómo lo medís?»

- nivel 1 no lo sabe
- nivel 2 da un número
- nivel 3 lo desglosa por tramos y sabe cuál domina

«¿qué pasa si una prueba falla de forma intermitente?»

- la respuesta mala: reintentarla
- la buena: es un defecto; se aísla, se arregla o se retira, porque una prueba en la que no se confía destruye el valor de todas

«¿cómo desplegáis un cambio de esquema?»

- discrimina mucho clase 260

«¿cuándo fue vuestra última vuelta atrás?»

- y si es «no hemos necesitado», es mala señal ley 22

y la que más revela

«¿cuánto tarda un equipo nuevo en tener su primer despliegue en producción?»

- mide el camino pavimentado, no la cadena

Y los diagnósticos típicos de una cadena mala:

SÍNTOMA

la cadena tarda 40 min

nadie confía en las pruebas
funciona en un entorno y no en otro

se despliega los viernes
nunca

el despliegue exige a una
persona concreta

y el ciclo es rápido pero
rompe mucho

CAUSA HABITUAL

pruebas mal ordenadas; lo
barato debe ir primero

intermitentes toleradas
el artefacto se reconstruye
por entorno

no hay vuelta atrás fiable
hay pasos manuales no escritos ley 30

falta verificación automática
tras desplegar

3. El modo de fracaso: el portero

Toda ruta tiene una forma característica de estropearse. La de esta es convertirse en quien autoriza.

CÓMO EMPIEZA, siempre con buena intención

un despliegue rompe producción

- «revisemos los despliegues antes»

otro rompe

- «que pase por nosotros»

y en seis meses

- el equipo de entrega aprueba todos los despliegues

QUÉ PRODUCE

cola de espera → lotes mayores → más riesgo

clase 260

responsabilidad desplazada

- «lo aprobasteis vosotros»

y el equipo pasa el día en trabajo repetitivo, sin
mejorar nada

- y las cuatro métricas empeoran a la vez

- es el mismo mecanismo del comité de cambios, con otro nombre clase 260

Y cómo se sale de ahí, que es trabajo de nivel 4:

- 1 MEDIR QUÉ FILTRA LA APROBACIÓN
de los últimos 200 despliegues revisados, ¿cuántos se rechazaron y cuántos incidentes causaron los aprobados?
→ CloudShop: 0,6 % rechazados y el 91 % de los incidentes venían de aprobados clase 260
- 2 SUSTITUIR CADA MOTIVO DE RECHAZO POR UNA COMPROBACIÓN AUTOMÁTICA
→ «no tenía vuelta atrás» → comprobación
→ «coincidía con otro cambio» → calendario
- 3 DEVOLVER LA RESPONSABILIDAD
quien escribe el cambio responde del despliegue
→ y para eso hace falta que PUEDA: señales, vuelta atrás, procedimiento y permisos
- 4 Y QUEDARSE CON EL CAMINO PAVIMENTADO
→ el equipo de entrega hace fácil lo correcto
→ no vigila que se haga

Y el segundo modo de fracaso, menos visible:

- LA CADENA COMO PRODUCTO PROPIO
se optimiza la cadena por elegancia interna
→ módulos, abstracciones, plantillas anidadas
→ y los equipos de producto no la entienden
- y entonces vuelven a pedir ayuda para cada cambio
→ y el equipo vuelve a ser cuello de botella, por otra vía
→ la medida correcta es la ADOPCIÓN y la autonomía, no la calidad interna de la cadena clase 267

4. El techo de la ruta

Dónde se acaba esta especialidad y qué la continúa.

- EL TECHO
se llega a tener una cadena rápida, fiable y adoptada
→ y las cuatro métricas dejan de mejorar porque el cuello ya no está ahí
- y lo que suele limitar entonces
la arquitectura del sistema
→ servicios acoplados que no se pueden desplegar por separado clases 106, 183
la organización
→ equipos que dependen unos de otros para entregar o la calidad del propio código
- y ninguna de las tres se arregla desde la cadena

Y las tres continuaciones naturales:

- a PLATAFORMA clase 267
la cadena es una parte del producto interno; el siguiente paso es todo lo demás que un equipo necesita para ir solo
- b FIABILIDAD clase 268
de «que el cambio llegue» a «que el sistema cumpla»
→ y añade presupuestos de error y decir que no
- c ARQUITECTURA clase 272
si lo que frena es el acoplamiento, el trabajo es rediseñar los límites

→ y la señal para elegir es qué te está frenando HOY

Y el consejo específico de esta ruta:

- APRENDE A MEDIR ANTES QUE A MONTAR
la mayoría llega con herramientas y sin números
→ y sin números no se puede demostrar mejora, ni justificar quitar una aprobación
- y la evidencia que vale en una entrevista

```

class 275

```

- sé cuáles son mis cuatro métricas y de dónde salen
- sé qué tramo domina mi tiempo de entrega
- mis pruebas permiten desplegar sin mirar; si no, lo sé
- no tolero pruebas intermitentes
- el artefacto se construye una vez y se promociona
- hay despliegue progresivo con vuelta atrás automática
- se ha ejecutado una vuelta atrás en los últimos 30 días
- los cambios de esquema se hacen en pasos reversibles
- mi equipo habilita y no aprueba
- mido adopción del camino pavimentado, no su elegancia
- un equipo nuevo llega a producción en días, y lo mido
- sé qué me está frenando ahora y en qué ruta se resuelve

y a los 9 meses

38 de 41

Y la conclusión que quedó escrita en el equipo:

la medida de esta ruta no es la calidad de la cadena
es cuántos equipos entregan solos y cuánto tardan
→ y eso es lo que la separa de un pasatiempo técnico

Caso 2 · El portero que se desmontó a sí mismo.

punto de partida

el equipo de entrega revisaba y ejecutaba todos los
despliegues a producción
2 personas dedicadas a ello
8 a 14 despliegues por semana

lo que se medía

tiempo de entrega	6,4 días
de los cuales, esperando al equipo	4,1 días
despliegues rechazados	0,6 %
incidentes por cambio	34/año
de despliegues que el equipo ejecutó	31

Y el argumento que preparó, que es el trabajo de nivel 4:

para cada motivo por el que alguna vez se había parado un
despliegue, una comprobación automática

motivo	sustituto
sin plan de vuelta atrás	comprobación en la cadena
coincide con otro cambio	calendario y bloqueo
	automático
falta la señal de salud	comprobación en la cadena
toca algo crítico	etiqueta que exige canario
	más lento
descripción incomprensible	plantilla obligatoria

→ 5 motivos, 5 comprobaciones

→ tiempo de implementación 3 semanas

Y la transición, que se hizo por fases para poder revertirla:

fase 1 las comprobaciones se ejecutan y el equipo sigue
revisando
→ 6 semanas en modo sombra clase 259
→ coincidencia entre la comprobación y la decisión
humana 194 de 197 casos
→ y los 3 discrepantes: la comprobación tenía razón
en 2

fase 2 tres equipos piloto despliegan solos
→ 5 semanas; incidentes por cambio: 0

fase 3 todos

Y las cifras a los 12 meses:

tiempo de entrega	6,4 días	3,2 h
despliegues por semana	8-14	47
tasa de fallo por cambio	1,8 %	0,6 %
tiempo de recuperación	47 min	11 min
personas dedicadas a ejecutar despliegues	2	0

Y lo que esas dos personas hicieron con su tiempo:

camino pavimentado para servicios nuevos
tiempo de un servicio nuevo hasta producción
3 semanas → 2 días
cadena de suministro: firma e inventario clase 216
y verificación automática tras desplegar
→ que es lo que llevó el tiempo de recuperación de
47 a 11 minutos

→ el mismo equipo, el mismo tamaño, otro efecto

→ y el trabajo repetitivo que hacían no era necesario:
era una consecuencia de no tener las comprobaciones

Y las preguntas de entrevista con las que se contrató a la siguiente persona.

- 1 «¿cuál es el tiempo de entrega de tu equipo y qué tramo lo domina?»
respuestas recibidas, 11 candidaturas
no lo sabe 6
da un número 4
lo desglosa por tramos 1
- 2 «tenéis una prueba que falla 1 de cada 20 veces sin motivo aparente. ¿Qué haces?»
reintentarla automáticamente 7
aislarla y arreglarla o retirarla 4
- 3 «¿cómo despliegas un cambio de esquema que ya escribió datos?»
lo describe en pasos reversibles 3
propone ventana de mantenimiento 6
no lo ha hecho 2
- 4 «¿cuándo fue la última vuelta atrás que ejecutasteis?»
una fecha concreta del último mes 2
«no hemos necesitado» 5
no lo sabe 4

→ la persona contratada fue la única que desglosó el tiempo de entrega por tramos, y también la única que dio fecha de la última vuelta atrás
→ y en el trabajo, esas dos respuestas predijeron bien

La lección que esta clase deja: la cadena técnicamente mejor tenía 3 de 41 servicios usándola, porque se había optimizado su calidad interna y no su adopción. Y el equipo que revisaba todos los despliegues rechazaba el 0,6 % mientras añadía 4,1 días de espera: al sustituir sus cinco motivos de rechazo por cinco comprobaciones automáticas, las cuatro métricas mejoraron a la vez y dos personas dejaron de hacer trabajo que nunca fue necesario.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-22-specializations-certifications-career/266-ruta-devops-y-delivery-engineer/lab.py
```

El laboratorio selecciona el motor de práctica decision y produce labresult.json. El escenario, sus comprobaciones y el artefacto esperado corresponden a esta clase; no requiere credenciales y deja explícito qué debe revalidarse en un sandbox real.

1. Lee exercise.steps y formula una predicción antes de ejecutar.
2. Ejecuta la práctica y verifica todos los elementos de checks.
3. Provoca el caso de negativetest y explica la señal observada.
4. Materializa devops-plan en evidence/ usando la plantilla indicada.
5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

El artefacto principal es una matriz de decisión ponderada y un ADR. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye devops-plan para el caso CloudShop. Incluye una alternativa descartada, un supuesto que pueda falsarse, una prueba de fallo y una decisión de rollback.

■ Criterio de aceptación

- [] lab.py termina con código 0 y genera JSON válido.

- [] La entrega conecta al menos tres requisitos con mecanismos verificables.
- [] Existe una prueba positiva y una prueba negativa con evidencia.
- [] Seguridad, costo y operación aparecen como decisiones, no como anexos.
- [] Se declara una limitación y una condición que obligaría a revisar el diseño.
- [] Otra persona puede repetir el recorrido sin conocimiento tácito.

■ Errores frecuentes

Síntoma	Causa probable	Corrección
La cadena es excelente y casi nadie la usa	Se optimizó la calidad interna en vez de la adopción y la autonomía	Mide cuántos equipos entregan solos y cuánto tardan; reduce configuración, tiempo y dependencias del equipo de plataforma.
El equipo de entrega revisa todos los despliegues y es el cuello de botella	Cada incidente añadió una aprobación en vez de una comprobación	Mide cuánto filtra la aprobación, sustituye cada motivo de rechazo por una comprobación automática y devuelve la responsabilidad con las herramientas para ejercerla.
Nadie confía en las pruebas y todos miran el despliegue	Se toleran pruebas intermitentes y se reintentan	Trata la intermitencia como defecto: aísla, arregla o retira; una prueba en la que no se confía destruye el valor de todas.
Funciona en un entorno y falla en el siguiente	El artefacto se reconstruye por entorno en vez de promocionarse	Construye una vez, firma y promociona el mismo artefacto; la configuración va fuera, no en la construcción.
No se despliega los viernes	No hay vuelta atrás fiable ni verificación automática tras desplegar	Automatiza la verificación y la vuelta atrás; si desplegar da miedo un día concreto, el problema no es el día.
Las cuatro métricas dejaron de mejorar pese al esfuerzo	El cuello de botella ya no está en la cadena sino en la arquitectura o en la organización	Identifica qué te frena hoy y cámbiate de ruta: plataforma, fiabilidad o arquitectura según el caso.

■ Seguridad, ética y costo

Trabaja con cuentas propias o sandboxes autorizados. No publiques secretos, identificadores reales ni datos personales. Antes de crear recursos pagos define presupuesto, etiquetas y comando de destrucción; después verifica que no queden recursos huérfanos. Los ejemplos locales enseñan contratos, pero no certifican cumplimiento ni disponibilidad de producción.

■ Preguntas de comprobación

1. ¿Cómo se sabe si el trabajo de esta ruta está produciendo efecto?
2. ¿Qué distingue una prueba que da confianza de una decorativa?
3. ¿Cómo empieza y cómo se sale del modo de fracaso del portero?
4. ¿Qué mide realmente la pregunta sobre el primer despliegue de un equipo nuevo?
5. ¿Cuál es el techo de esta ruta y qué la continúa?

Referencias

- Humble, J. y Farley, D. (2010). Continuous Delivery. <<https://www.pearson.com/en-us/subject-catalog/p/continuous-delivery/P200000009415>>
- Forsgren, N., Humble, J. y Kim, G. (2018). Accelerate. <<https://itrevolution.com/product/accelerate/>>
- Google (2024). DORA capabilities: continuous delivery. <<https://dora.dev/capabilities/>>
- OpenSSF (2024). SLSA: supply chain levels for software artifacts. <<https://slsa.dev/>>

- Skelton, M. y Pais, M. (2019). Team Topologies — equipos habilitadores frente a porteros.
<<https://teamentopologies.com/book>>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 22 en PDF · Manual integral

Anterior	Índice	Siguiente
← 265 · Ruta Cloud Engineer y mapa de competencias	Parte 22 · Programa	267 · Ruta Platform Engineer →

Evaluación — 266 Ruta DevOps y Delivery Engineer

Preguntas

1. Explica el problema que resuelve esta capacidad sin mencionar un proveedor.
2. Identifica una frontera de responsabilidad y su propietario.
3. Compara dos alternativas mediante confiabilidad, seguridad, costo y operación.
4. ¿Qué demuestra el laboratorio y qué no puede demostrar?
5. Propón un caso límite o una prueba de fallo.

Reto verificable

Entrega devops-plan con diagrama o configuración, evidencia de ejecución, alternativa descartada, riesgo residual y criterio de rollback.

Criterio de aceptación

- [] Resultado reproducible por una segunda persona.
- [] Evidencia vinculada a cada afirmación técnica.
- [] Prueba negativa o de fallo incluida.
- [] Costos y permisos expresados con unidades y alcance.
- [] Limitaciones declaradas sin presentar la simulación como producción.

Escala

Nivel	Evidencia
A — competente	Cumple todo y defiende trade-offs con datos.
B — en desarrollo	Cumple lo esencial; falta profundidad en una dimensión.
C — inicial	Artefacto parcial o conclusión sin evidencia suficiente.
0	Sin entrega reproducible.

267 — Ruta Platform Engineer

Parte: 22 — Especializaciones, certificaciones y práctica profesional Nivel: intermedio-avanzado · Horas estimadas: 4
Laboratorio: decision · Estado: EXECUTABLECORE

Propósito

La ruta de plataforma: tratar la infraestructura interna como un producto con usuarios que pueden no usarlo. La clase da lo que la distingue de la ruta de entrega, la única métrica que importa —adopción voluntaria—, el criterio para decidir qué entra en la plataforma y qué no, y su modo de fracaso: construir la plataforma que el equipo quiere construir, no la que los equipos necesitan.

Resultados de aprendizaje

Al finalizar podrás:

1. Definir la plataforma como producto con usuarios y competencia.
2. Medir adopción voluntaria, autonomía y tiempo hasta el primer valor.
3. Decidir qué entra en la plataforma y qué se deja fuera.
4. Evitar construir para el equipo de plataforma en vez de para sus usuarios.
5. Reconocer cuándo una organización todavía no necesita plataforma.

Conceptos centrales

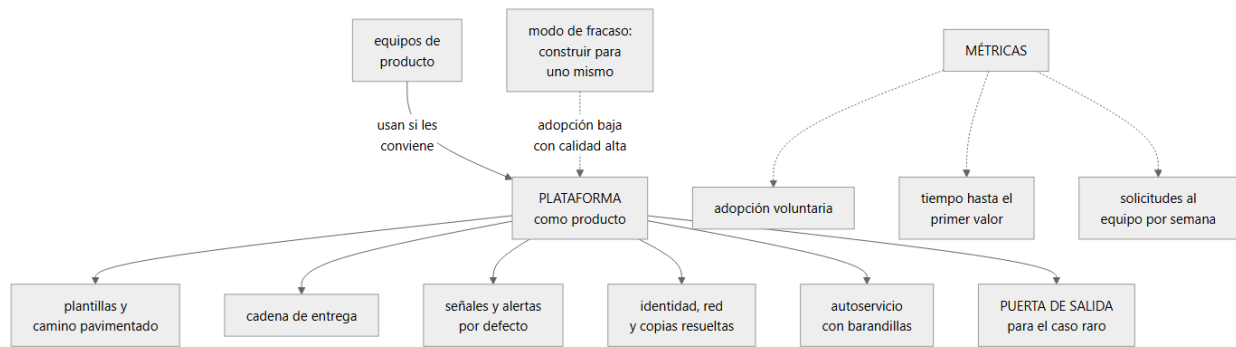
Concepto	Comprensión verificable
plataforma interna	Producto que otros equipos usan para entregar software sin depender de personas concretas.
adopción voluntaria	Porcentaje de equipos que la usan pudiendo no hacerlo. La medida honesta.
camino pavimentado	La forma fácil de hacer lo correcto. La plataforma es su implementación.
puerta de salida	Posibilidad de salirse de la plataforma para un caso concreto sin salirse del todo.
tiempo hasta el primer valor	Cuánto tarda un equipo nuevo desde que empieza hasta que sirve algo en producción.
plataforma prematura	Producto interno construido antes de que existan varios equipos con problemas comunes.

Modelo mental

Una especialización combina fundamentos, evidencia de proyectos y juicio bajo restricciones; una insignia sin práctica no sustituye esa combinación.

Aplicado a esta clase, separa siempre cuatro planos: intención (qué necesita el usuario), configuración (qué declaramos), estado observado (qué existe de verdad) y evidencia (cómo sabemos que cumple). Confundirlos produce diseños que se ven correctos en un diagrama pero fallan al operar.

■ Flujo de razonamiento



Desarrollo

1. Producto, no infraestructura

La diferencia entre esta ruta y las demás no es técnica: es que aquí los usuarios son internos y pueden no usarte.

LO QUE CAMBIA AL TRATARLO COMO PRODUCTO

- hay usuarios, y se les pregunta
- hay alternativas, y compites con ellas
 - la alternativa siempre existe: hacerlo a mano
- hay adopción, y se mide
- hay versiones, migraciones y retirada
- y hay soporte

- y si la plataforma es obligatoria, todo esto se apaga
- la obligatoriedad esconde los defectos en vez de corregirlos

ley 16

Y lo que distingue esta ruta de la de entrega:

ENTREGA	PLATAFORMA
el camino del cambio a producción	todo lo que un equipo necesita para ir solo
las cuatro métricas	adopción y autonomía
una cadena	un producto con muchas piezas: cadena, plantillas, señales, identidad, red, datos, coste

- la de entrega es una PARTE de la de plataforma
- y quien viene de entrega suele traer la cadena resuelta y la conversación con usuarios sin empezar

Y las tres métricas que dicen la verdad:

- 1 ADOPCIÓN VOLUNTARIA
 - equipos que la usan pudiendo no hacerlo
 - y si es obligatoria, la métrica no existe y hay que preguntar: «¿la usarías si no fuera obligatoria?»
 - 2 TIEMPO HASTA EL PRIMER VALOR
 - de un equipo nuevo a servir algo en producción
 - semanas significa que la plataforma no reduce trabajo
 - días significa que sí
 - 3 SOLICITUDES AL EQUIPO DE PLATAFORMA POR SEMANA
 - y su tendencia
 - si crece con el número de equipos, no hay autoservicio: hay un mostrador
- y una cuarta, para el equipo
- qué porcentaje del tiempo va a atender solicitudes
 - por encima del 40 %, la plataforma no avanza

Y la pregunta que ordena las prioridades:

«¿QUÉ ES LO QUE MÁS VECES HAN TENIDO QUE RESOLVER LOS

EQUIPOS POR SU CUENTA ESTE TRIMESTRE?»

- eso es la siguiente pieza
- y no lo que el equipo de plataforma encuentra interesante

2. Qué entra y qué no

El criterio de alcance decide si la plataforma es útil o es una carga.

ENTRA SI

- varios equipos lo necesitan
- hacerlo mal tiene consecuencias serias
 - identidad, red, copias, secretos, señales
- y resolverlo bien exige conocimiento que no todos tienen

NO ENTRA SI

- lo necesita un solo equipo
 - que lo haga él; y si aparece un segundo, se reconsidera
- varía mucho entre equipos
 - la abstracción costará más de lo que ahorra
 - o está cambiando rápido
 - congelarlo en una plantilla lo hace obsoleto
- y la regla práctica: se abstrae al TERCER caso, no al primero
- dos casos no dicen dónde está la variación

Y la pieza que casi nadie diseña y que decide la adopción:

LA PUERTA DE SALIDA

«quiero todo lo de la plataforma, salvo el balanceador, que en mi caso necesita algo que no cubrés»

si la respuesta es «entonces sal del todo»

- el equipo sale del todo
- y se lleva consigo las señales, las copias y los permisos bien puestos

si la respuesta es «cambia esa pieza y quédate con el resto»

- el equipo se queda

- una plataforma sin puerta de salida pierde a sus usuarios más avanzados
- y esos son los que descubren lo que le falta

Y los niveles de la ruta:

NIVEL 2 · RESUELVO

- monta piezas de la plataforma que funcionan
- documenta y da soporte
- y sabe por qué un equipo no la usa, porque lo ha preguntado

NIVEL 3 · DISEÑO

- decide el alcance: qué entra y qué se deja fuera
- diseña interfaces estables y migraciones sin ruptura
- clase 106
- y mide adopción, no calidad interna

NIVEL 4 · CAMBIO EL SISTEMA

- la plataforma cambia cómo se organiza la empresa
- los equipos de producto responden de sus servicios porque PUEDEN
- y el equipo de plataforma deja de aparecer en la ruta crítica de nadie

Y el requisito que la hace sostenible:

LA PLATAFORMA TAMBIÉN SE OPERA

- tiene guardia, señales, procedimientos y ensayos
- parte 21
- y su indisponibilidad afecta a todos los equipos

- tiene compromiso de servicio con sus usuarios
→ «la cadena de entrega estará disponible el 99,5 %»
→ y si no se cumple, los equipos vuelven a hacerlo a
mano

3. El modo de fracaso y la plataforma prematura

Dos formas de equivocarse, y la segunda es más común de lo que parece.

MODOS DE FRACASO 1 · CONSTRUIR PARA UNO MISMO

el equipo construye lo que le parece elegante
→ abstracciones profundas, configuración expresiva,
plantillas anidadas
y los usuarios no lo entienden

la señal

- calidad interna alta y adopción baja
- y el equipo lo interpreta como falta de formación
- cuando es una respuesta del mercado interno

la corrección

hablar con usuarios cada semana, no cada trimestre
medir el tiempo hasta el primer valor con alguien
real, mirando
y aceptar que la configuración sobra casi siempre

MODO DE FRACASO 2 · LA PLATAFORMA PREMATURA

- se monta una plataforma con dos equipos y tres servicios
- y no hay suficientes casos para saber qué abstraer
- y se congela una forma de trabajar que aún no está entendida

la señal

la plataforma cambia cada mes por peticiones
y el equipo de plataforma es más grande que los
equipos que la usan

la corrección

no montar plataforma hasta que haya problemas
REPETIDOS entre varios equipos
→ y mientras tanto, hacer bibliotecas y plantillas
copiables, que no crean dependencia

Y el techo de la ruta:

EL TECHO

la plataforma funciona, se adopta y el equipo ya no está en la ruta crítica de nadie
→ y entonces la limitación pasa a ser la arquitectura de los sistemas o la organización

continuaciones

- | | | |
|---|--|-----------|
| a | ARQUITECTURA | clase 272 |
| | los límites entre servicios y equipos | |
| b | FIABILIDAD | clase 268 |
| | responder de propiedades del conjunto | |
| c | o dirección técnica | |
| | donde el trabajo es de personas y de prioridades | |

Y la señal de que la organización todavía no necesita esta ruta:

- hay menos de tres equipos entregando software
 - los problemas que se repiten aún no se han repetido
 - no hay nadie que pueda dedicarse a ello de forma sostenida
- si se cumple alguna, hacer plantillas y bibliotecas, no plataforma

Y la lista de comprobación de la clase:

- la plataforma tiene usuarios y se les pregunta
- la adopción es voluntaria, o sé qué esconde que no lo sea

- mido tiempo hasta el primer valor con alguien real
- las solicitudes al equipo no crecen con el número de equipos
- el equipo dedica menos del 40 % a atender solicitudes
- se abstrae al tercer caso, no al primero
- existe puerta de salida por pieza
- hay interfaz estable y migraciones sin ruptura
- la plataforma se opera: guardia, señales, procedimientos
- tiene compromiso de servicio con sus usuarios
- sé por qué cada equipo que no la usa no la usa
- y sé si la organización necesita plataforma o todavía no

Y el cierre que enlaza con la clase siguiente: la plataforma hace que otros puedan; queda quien responde de que el sistema cumpla lo prometido, con un número acordado y la autoridad para frenar. La ruta de fiabilidad es la materia de la clase 268.

4. Cómo se demuestra esta ruta

Lo que vale como evidencia, en entrevista y en revisión de desempeño.

LO QUE NO VALE

- «monté la plataforma interna»
- «usamos Kubernetes y plantillas»
- describe el qué y no el efecto

LO QUE VALE

- «la adopción pasó de 3 a 38 de 41 servicios en 9 meses, y lo que lo movió fue bajar las pruebas de 40 a 9 minutos y la configuración de 11 ficheros a 1»
- «el tiempo de un servicio nuevo hasta producción pasó de 3 semanas a 2 días»
- «las solicitudes al equipo bajaron de 34 a 6 por semana con el doble de equipos»

→ efecto, mecanismo y cifra clase 275

Y las preguntas que separan niveles:

- «¿cuántos equipos la usan y cuántos podrían no usarla?»
- si es obligatoria, ¿qué pasaría si dejara de serlo?

- «¿qué pieza de la plataforma retirasteis y cómo?»
- nivel 3 ha retirado algo; nivel 2 solo ha añadido

- «¿qué hace un equipo que necesita algo que no cubrís?»
- la respuesta revela si hay puerta de salida

- «¿qué porcentaje del tiempo de tu equipo va a soporte?»
- y su tendencia

y la que más discrimina

- «¿por qué el último equipo que no la adoptó no la adoptó?»
- quien no lo sabe no está tratando esto como producto

Y el consejo específico de la ruta:

SIÉNTATE CON UN EQUIPO NUEVO Y MIRA SIN AYUDAR

- con un cronómetro y sin decir nada
- dónde se atasca, qué busca y no encuentra, qué interpreta al revés

- una hora de esto vale más que un trimestre de encuestas
- y es el mismo método de los ensayos: observar sin intervenir clase 261

Y el error de comunicación que más cuesta:

PRESENTAR LA PLATAFORMA COMO AHORRO DE COSTE

- y entonces se evalúa contra el coste del equipo que la mantiene, y siempre pierde

PRESENTARLA COMO CAPACIDAD DE ENTREGA

- «los equipos entregan en 2 días lo que antes tardaba

3 semanas, con 14 equipos en vez de 3»
→ y eso sí se compara con lo que produce

Ejemplo trabajado

La plataforma de CloudShop, tres años en cifras. Lo que sigue es el intento prematuro que se desmontó, la reconstrucción con puerta de salida, y las dos preguntas que decidieron cada prioridad.

Intento 1 · La plataforma prematura.

momento

2 equipos de producto, 6 servicios
1 persona dedicada a plataforma

lo que se construyó, en 5 meses

abstracción propia sobre la infraestructura
un lenguaje de descripción de servicios
un panel interno de despliegue
y plantillas para «cualquier servicio»

lo que pasó

cada servicio nuevo requería extender la abstracción
→ 11 extensiones en 5 meses
la abstracción tenía casos especiales para 4 de los 6 servicios
y la persona de plataforma pasó a ser dependencia de todo

y la señal clara

tiempo de un servicio nuevo
antes de la plataforma 4 días
con la plataforma 6 días

Y el diagnóstico:

con 6 servicios no había forma de saber qué variaba y qué no

→ se abstraigo al primer caso, no al tercero
→ y la abstracción codificó las particularidades de los dos primeros servicios

lo que se hizo

se retiró la abstracción
se dejaron PLANTILLAS COPIABLES
→ cada equipo copia y modifica
→ duplicación, sí; dependencia, no
y se apuntó lo que se repetía

→ 14 meses después, con 9 equipos y 27 servicios, esas notas dijeron exactamente qué abstraer

Intento 2 · La reconstrucción.

punto de partida

9 equipos, 27 servicios
y una lista de lo que cada equipo había resuelto por su cuenta, recogida durante 14 meses

lo que se repetía, por número de equipos que lo habían resuelto solos

señales y paneles básicos	9/9
identidad y permisos de servicio	9/9
copias y su verificación	8/9
cadena de entrega	9/9
gestión de secretos	7/9
red y exposición	9/9
alta disponibilidad entre zonas	6/9
colas y reintentos	5/9
caché	4/9
búsqueda	2/9

→ se construyó de arriba abajo

→ y búsqueda y caché se dejaron fuera, con nota

Y la puerta de salida, que fue la decisión de diseño más discutida:

cada pieza se puede sustituir sin salir de la plataforma
→ «uso vuestra cadena, vuestras señales y vuestros permisos, pero mi red la monto yo»
→ y la plataforma sigue dando lo demás

coste de esta decisión
las piezas tienen que tener interfaz, no estar
entrelazadas clase 106
→ más trabajo de diseño

y lo que produjo	
equipos que usan la plataforma completa	31
equipos que sustituyen 1 o 2 piezas	7
equipos fuera del todo	3

→ sin puerta de salida, esos 7 habrían salido del todo
→ y con ellos, 7 servicios sin señales ni copias bien
puestas

Las dos preguntas que decidieron cada prioridad.

PREGUNTA 1, cada trimestre, a cada equipo
«¿qué has tenido que resolver por tu cuenta este trimestre?»

PREGUNTA 2, en la sesión de observación se sienta un equipo nuevo con la plataforma y el equipo de plataforma MIRA sin ayudar, una hora

lo que la observación encontró, y las encuestas no
el ejemplo de la documentación no funcionaba
(una variable había cambiado de nombre)
el mensaje de error al faltar un permiso decía
«acceso denegado» sin decir cuál
la plantilla pedía elegir entre tres opciones sin
explicar la diferencia
y el primer despliegue fallaba siempre por un valor por
defecto que nadie cambiaba

- los cuatro son triviales de arreglar
- y los cuatro costaban entre 20 y 90 minutos a CADA equipo nuevo
- ninguna encuesta los había mencionado

Las cifras, a los tres años.

equipos de producto	2	9	14
servicios	6	27	41
personas en plataforma	1	3	4

adopción voluntaria	-	18/27	38/41
tiempo hasta el primer valor	6 días	5 días	2 días
solicitudes al equipo/semana	-	34	6
% del tiempo en soporte	-	61 %	18 %
disponibilidad de la cadena	-	97,1 %	99,7 %

servicios con señales por defecto	-	11/27	41/41
servicios con copias verificadas	-	9/27	39/41
servicios con permisos mínimos	-	6/27	38/41

Y la cifra que el equipo usó para defender su presupuesto:

no se presentó como ahorro de coste
se presentó así

equipos que entregan solos	2 → 14
servicios en producción	6 → 41
personas de plataforma	1 → 4

servicios por persona de plataforma 6 → 10,25
y tiempo de un servicio nuevo 4 días → 2 días

→ el sistema creció 7 veces con 4 veces el equipo

→ y la calidad por defecto subió: 41 de 41 con señales,
frente a 11 de 27 dos años antes

Y el fracaso que quedó registrado:

la pieza de colas y reintentos se construyó en el año 2
la habían resuelto solos 5 de 9 equipos
adopción a los 6 meses 2 de 14

por qué
cada equipo tenía necesidades distintas de orden,
reintento y duplicados clase 210
→ la variación era el problema, no la repetición
→ «lo han resuelto muchos» no implica «lo resuelven
igual»

se retiró
y se sustituyó por documentación y ejemplos revisados
→ adopción de los ejemplos 11 de 14

→ y la lección quedó: antes de abstraer, comprobar que
las N soluciones se parecen

La lección que esta clase deja: la primera plataforma hizo que un servicio nuevo tardara 6 días en vez de 4, porque se abstrajo con seis servicios y la abstracción codificó las particularidades de los dos primeros. Y la puerta de salida —la decisión más discutida— retuvo a 7 equipos que habrían salido del todo, y con ellos siete servicios que habrían perdido señales, copias y permisos mínimos.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-22-specializations-certifications-career/267-ruta-platform-engineer/lab.py
```

El laboratorio selecciona el motor de práctica decision y produce labresult.json. El escenario, sus comprobaciones y el artefacto esperado corresponden a esta clase; no requiere credenciales y deja explícito qué debe revalidarse en un sandbox real.

1. Lee exercise.steps y formula una predicción antes de ejecutar.
2. Ejecuta la práctica y verifica todos los elementos de checks.
3. Provoca el caso de negativetest y explica la señal observada.
4. Materializa platform-engineer-plan en evidence/ usando la plantilla indicada.
5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

El artefacto principal es una matriz de decisión ponderada y un ADR. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye platform-engineer-plan para el caso CloudShop. Incluye una alternativa descartada, un supuesto que pueda falsarse, una prueba de fallo y una decisión de rollback.

■ Criterio de aceptación

- [] lab.py termina con código 0 y genera JSON válido.
- [] La entrega conecta al menos tres requisitos con mecanismos verificables.
- [] Existe una prueba positiva y una prueba negativa con evidencia.
- [] Seguridad, costo y operación aparecen como decisiones, no como anexos.
- [] Se declara una limitación y una condición que obligaría a revisar el diseño.
- [] Otra persona puede repetir el recorrido sin conocimiento tácito.

■ Errores frecuentes

Síntoma	Causa probable	Corrección
La plataforma tiene calidad alta y adopción baja	Se construyó para el equipo de plataforma, no para sus usuarios	Siéntate a mirar a un equipo nuevo sin ayudar durante una hora; encontrarás obstáculos triviales que ninguna encuesta menciona.
Cada servicio nuevo obliga a extender la abstracción	Se abstraigo demasiado pronto, con muy pocos casos para saber qué varía	Abstrae al tercer caso; hasta entonces usa plantillas copiables, que duplican pero no crean dependencia.
Un equipo con una necesidad especial abandona la plataforma entera	No hay puerta de salida por pieza	Diseña piezas con interfaz para que se puedan sustituir una a una; sin eso pierdes a los usuarios más avanzados.
Las solicitudes al equipo crecen con cada equipo nuevo	Hay mostrador en vez de autoservicio	Convierte en autoservicio con validación cada solicitud repetida y vigila que el soporte no pase del 40 % del tiempo.
Se construyó una pieza que muchos habían resuelto y casi nadie la adopta	Se confundió repetición con uniformidad: cada equipo lo resolvía distinto	Antes de abstraer comprueba que las soluciones existentes se parecen; si varían mucho, documenta y da ejemplos.
La plataforma se justifica como ahorro y siempre pierde en el presupuesto	Presentada como coste, se compara con el coste del equipo que la mantiene	Preséntala como capacidad de entrega: equipos que entregan solos, servicios por persona de plataforma y tiempo hasta producción.

■ Seguridad, ética y costo

Trabaja con cuentas propias o sandboxes autorizados. No publiques secretos, identificadores reales ni datos personales. Antes de crear recursos pagos define presupuesto, etiquetas y comando de destrucción; después verifica que no queden recursos huérfanos. Los ejemplos locales enseñan contratos, pero no certifican cumplimiento ni disponibilidad de producción.

■ Preguntas de comprobación

1. ¿Qué cambia al tratar la infraestructura interna como producto?
2. ¿Cuáles son las tres métricas que dicen la verdad sobre una plataforma?
3. ¿Qué criterio decide si algo entra en la plataforma?
4. ¿Para qué sirve la puerta de salida y qué pasa sin ella?
5. ¿Cómo se reconoce una plataforma prematura y qué se hace en su lugar?

Referencias

- Skelton, M. y Pais, M. (2019). Team Topologies — equipos de plataforma como habilitadores. <<https://teamtopologies.com/book>>
- Bottcher, E. (2018). What I talk about when I talk about platforms. <<https://martinfowler.com/articles/talk-about-platforms.html>>
- Google (2024). DORA: platform engineering and developer experience. <<https://dora.dev/research/>>
- CNCF (2024). Platform engineering maturity model. <<https://tag-app-delivery.cncf.io/whitepapers/platform-eng-maturity-model/>>
- Humanitec (2024). Platform as a product: reference architectures. <<https://humanitec.com/reference-architectures>>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 22 en PDF · Manual integral

Anterior	Índice	Siguiente
← 266 · Ruta DevOps y Delivery Engineer	Parte 22 · Programa	268 · Ruta Site Reliability Engineer →

Evaluación — 267 Ruta Platform Engineer

Preguntas

1. Explica el problema que resuelve esta capacidad sin mencionar un proveedor.
2. Identifica una frontera de responsabilidad y su propietario.
3. Compara dos alternativas mediante confiabilidad, seguridad, costo y operación.
4. ¿Qué demuestra el laboratorio y qué no puede demostrar?
5. Propón un caso límite o una prueba de fallo.

Reto verificable

Entrega platform-engineer-plan con diagrama o configuración, evidencia de ejecución, alternativa descartada, riesgo residual y criterio de rollback.

Criterio de aceptación

- ☐ Resultado reproducible por una segunda persona.
- ☐ Evidencia vinculada a cada afirmación técnica.
- ☐ Prueba negativa o de fallo incluida.
- ☐ Costos y permisos expresados con unidades y alcance.
- ☐ Limitaciones declaradas sin presentar la simulación como producción.

Escala

Nivel	Evidencia
A — competente	Cumple todo y defiende trade-offs con datos.
B — en desarrollo	Cumple lo esencial; falta profundidad en una dimensión.
C — inicial	Artefacto parcial o conclusión sin evidencia suficiente.
0	Sin entrega reproducible.

268 — Ruta Site Reliability Engineer

Parte: 22 — Especializaciones, certificaciones y práctica profesional Nivel: intermedio-avanzado · Horas estimadas: 4
Laboratorio: decision · Estado: EXECUTABLECORE

Propósito

La ruta de fiabilidad: responder de que el sistema cumpla lo prometido, con un número acordado y la autoridad para actuar sobre él. La clase da lo que la distingue de «operar bien» —el objetivo negociado, el presupuesto de error y la potestad de frenar—, las competencias que se miden, y su modo de fracaso: convertirse en el equipo que hace las guardias de los demás.

Resultados de aprendizaje

Al finalizar podrás:

1. Definir objetivos de nivel de servicio que representen al usuario.
2. Usar el presupuesto de error como mecanismo de decisión, no de informe.
3. Negociar el número con negocio y con producto, con sus costes.
4. Evitar convertirse en el equipo que absorbe la guardia ajena.
5. Reconocer el techo de la ruta y qué la continúa.

Conceptos centrales

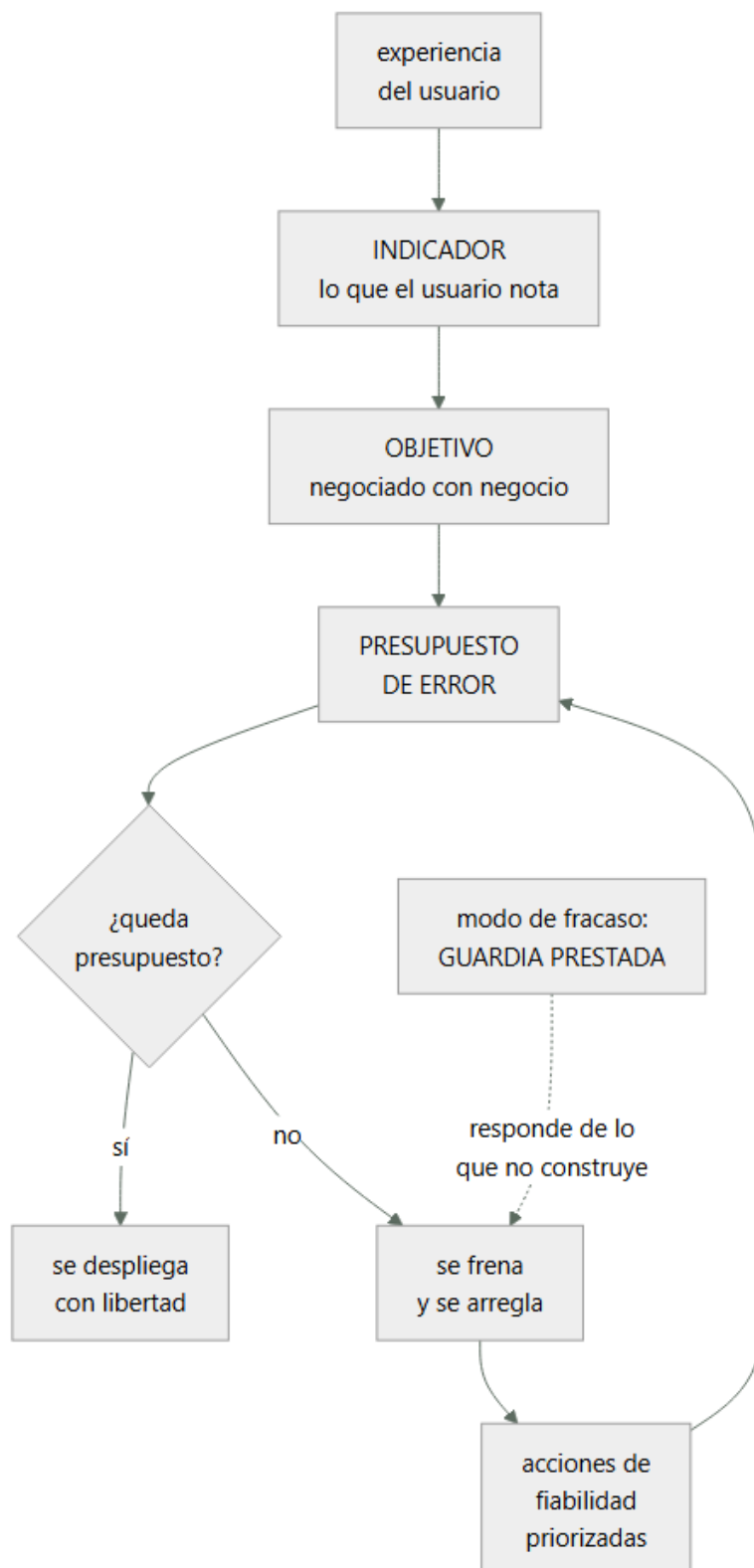
Concepto	Comprensión verificable
indicador de nivel de servicio	Medida de algo que el usuario experimenta: éxito de una operación, latencia, frescura.
objetivo de nivel de servicio	El valor acordado que ese indicador debe cumplir. Se negocia, no se declara.
presupuesto de error	Lo que sobra del objetivo. Se gasta desplegando; agotarlo tiene consecuencias acordadas.
potestad de frenar	Autoridad para detener despliegues cuando el presupuesto se agota. Sin ella, el objetivo es decorativo.
trabajo repetitivo	Tarea manual sin valor duradero que crece con el sistema. Su límite define esta ruta.
guardia prestada	Modo de fracaso en que el equipo de fiabilidad responde de servicios que no construye.

Modelo mental

Una especialización combina fundamentos, evidencia de proyectos y juicio bajo restricciones; una insignia sin práctica no sustituye esa combinación.

Aplicado a esta clase, separa siempre cuatro planos: intención (qué necesita el usuario), configuración (qué declaramos), estado observado (qué existe de verdad) y evidencia (cómo sabemos que cumple). Confundirlos produce diseños que se ven correctos en un diagrama pero fallan al operar.

■ Flujo de razonamiento



Desarrollo

1. Qué distingue esta ruta

No es «operar bien»: eso lo hace la parte 21 entera. Lo que define esta ruta son tres cosas que casi nunca están juntas.

- 1 UN NÚMERO ACORDADO
no «que vaya bien», sino «el 99,9 % de las compras se

completan en menos de 2 segundos»
→ acordado con negocio, no declarado por ingeniería

2 UN MECANISMO QUE CONVIERTE EL NÚMERO EN DECISIONES
el presupuesto de error

- mientras quede, se despliega con libertad
- cuando se agota, se para y se arregla

3 Y LA POTESTAD DE APLICARLO

- sin esto, el objetivo es un informe mensual que nadie lee
- y esta es la parte que no es técnica y la que decide si la ruta existe de verdad

Y el indicador, que es donde se cometen los errores:

UN BUEN INDICADOR

mide lo que el usuario experimenta
se mide donde el usuario está, no en el servidor
tiene un umbral claro de bueno y malo
y su caída significa que alguien está sufriendo

UN MAL INDICADOR

disponibilidad de la máquina
uso de CPU
«la petición devolvió 200»
→ aunque el contenido esté vacío ley 29
y el promedio de latencia
→ la media oculta la cola clase 186

- la prueba: «si esto empeora, ¿lo nota alguien?»
- y la inversa, que es la que más falla:
«si algo empeora para el usuario, ¿esto lo refleja?»

Y cómo se elige el objetivo, que se negocia:

NO SE ELIGE «EL MÁS ALTO POSIBLE»

cada nueve adicional multiplica el coste

99,9 %	8,7 h al año de margen
99,95 %	4,4 h
99,99 %	52 min

→ y pasar de 99,9 a 99,99 puede costar más que todo el servicio

SF FL TGF MTRANDO

qué nota el usuario de verdad
→ si el cliente móvil ya falla el 0,4 % por la red,
un 99,99 % de servidor no lo nota nadie
qué prometen los contratos
qué cuesta cada nivel
y qué disponibilidad tienen las dependencias

clase 185

- y el objetivo debe ser PEOR que lo que hoy se consigue por casualidad
- si el objetivo se cumple siempre sin esfuerzo, no está midiendo nada útil

2. El presupuesto de error como mecanismo

El presupuesto de error solo sirve si tiene consecuencias acordadas de antemano. Si no, es una métrica más.

CÓMO FUNCIONA

objetivo 99,9 % → presupuesto 0,1 %
en 30 días, unos 43 minutos de fallo total equivalente

cada incidente y cada despliegue fallido lo consumen y se repone al empezar la ventana siguiente

LAS CONSECUENCIAS, escritas ANTES

presupuesto > 50 %	se despliega con libertad; se pueden asumir riesgos
presupuesto < 25 %	solo cambios de bajo riesgo; revisión más estricta
presupuesto agotado	se congelan las funciones

nuevas; el equipo trabaja en
fiabilidad hasta reponerlo

- y esto último es lo que le da poder
- y es lo que hay que acordar con producto ANTES de que ocurra, no durante

Y lo que el mecanismo consigue, que es su verdadero propósito:

CONVIERTE UNA DISCUSIÓN DE OPINIONES EN UNA DE DATOS
antes «hay que ir más despacio» / «hay que entregar»
ahora «queda el 18 % del presupuesto»

y alinea incentivos
a producto le interesa que el sistema sea fiable, porque
si no, se le congelan las funciones
a ingeniería le interesa no ser conservadora de más,
porque el presupuesto sin gastar es margen
desaprovechado

- y ese segundo punto sorprende: gastar POCO presupuesto también es una señal
- significa que se podría entregar más rápido o que el objetivo está mal puesto

Y los errores más comunes con el presupuesto:

- 1 NO TENER CONSECUENCIAS
se informa y no pasa nada
→ y entonces es un panel bonito
- 2 PONER EL OBJETIVO DONDE YA SE CUMPLE
→ presupuesto siempre lleno, nunca decide nada
- 3 MEDIRLO SOBRE LA VENTANA EQUIVOCADA
30 días móviles funciona bien
→ trimestral esconde; semanal es demasiado ruidoso
- 4 EXCLUIR INCIDENTES «QUE NO CUENTAN»
→ «esto fue del proveedor»
→ el usuario no distingue de quién fue clase 185
→ y si se excluye, el número deja de representarlo
- 5 Y TENER DEMASIADOS
→ 3 a 5 objetivos por servicio; más no se gestionan

3. Competencias, niveles y el trabajo repetitivo

Lo que se mide en esta ruta, por nivel.

NIVEL 2 · RESUELVO
define indicadores que representan al usuario
instrumenta y mide con corrección clases 211, 257
diagnostica y arregla clase 258
escribe procedimientos y análisis posteriores
clases 111, 259
y responde de una guardia

NIVEL 3 · DISEÑO
negocia objetivos con negocio y sostiene el número
diseña degradación y vertido de carga clase 262
hace análisis de modos de fallo antes de que ocurran
dimensiona capacidad y márgenes
y dice que no a un despliegue, con argumento

NIVEL 4 · CAMBIO EL SISTEMA
el presupuesto de error se respeta sin discusión porque
la organización lo acordó
la fiabilidad se diseña en los servicios nuevos, no se
añade después
y el trabajo repetitivo baja mientras el sistema crece

Y el límite que define la salud de la ruta:

EL LÍMITE AL TRABAJO REPETITIVO

la práctica clásica: no más del 50 % del tiempo
→ y el resto, en trabajo que reduce el futuro trabajo

y si se pasa
se devuelve la guardia al equipo que construye
o se para de aceptar servicios nuevos

→ este límite es lo que impide el modo de fracaso
→ y hay que escribirlo antes de necesitarlo

Y el modo de fracaso, que es el más frecuente de todas las rutas:

LA GUARDIA PRESTADA

el equipo de fiabilidad acaba respondiendo de servicios
que no construye

cómo empieza

«vosotros sabéis operar mejor»
y el equipo de producto deja de recibir alertas

qué produce

quien construye no sufre lo que construye
→ y entonces no lo arregla clase 111
el equipo de fiabilidad se llena de trabajo repetitivo
y su conocimiento de cada servicio es superficial

cómo se evita

CRITERIOS DE ENTRADA para aceptar un servicio
señales definidas y objetivos acordados
procedimientos ejecutables clase 259
despliegue progresivo y vuelta atrás clase 260
y el equipo que lo construye COMPARTE la guardia
y criterios de SALIDA
→ si el servicio deja de cumplirlos, se devuelve

→ y devolver un servicio es la acción más impopular y más
necesaria de esta ruta

4. Negociar el número y el techo de la ruta

La parte que hace difícil esta especialidad no es medir: es sostener el número cuando incomoda.

CÓMO SE NEGOCIA CON NEGOCIO

no se pregunta «¿qué disponibilidad quieres?»
→ la respuesta siempre es «la máxima»

se pregunta

«¿cuánto vale para el negocio una hora de caída?»
«¿qué prefieres: esta función un mes antes o pasar de
99,9 a 99,95?»
«¿qué le pasa a un usuario cuando esto falla?»

y se presenta el coste de cada nivel

→ con la cifra de la parte 19: la redundancia entre
regiones no es un ajuste, es otro sistema
clase 187

Y cómo se sostiene:

CUANDO EL PRESUPUESTO SE AGOTA Y HAY UN LANZAMIENTO

la conversación no es «¿lo paramos?»
es «acordamos esto; ¿queréis cambiar el acuerdo?»
→ y cambiar el acuerdo es legítimo, si es explícito
→ lo ilegítimo es ignorarlo

y la salida honesta

«podemos lanzarlo si asumimos que el objetivo baja a
99,5 % este trimestre; ¿lo asumimos?»
→ decisión de negocio, registrada

→ el trabajo de esta ruta no es impedir: es hacer la
decisión VISIBLE

Y el techo:

EL TECHO

el sistema cumple, el presupuesto se respeta y el trabajo repetitivo está bajo control
→ y lo que limita entonces es la arquitectura o la organización

continuaciones

- a ARQUITECTURA clase 272
si el límite es cómo está construido
- b PLATAFORMA clase 267
si el límite es que cada equipo lo resuelve solo
- c o dirección técnica
si el límite es cómo se priorizan las decisiones

Y la lista de comprobación de la clase:

- mis indicadores miden lo que el usuario experimenta
- se miden donde el usuario está
- el objetivo está acordado con negocio, no declarado
- el objetivo no se cumple solo, por casualidad
- hay entre 3 y 5 objetivos por servicio
- el presupuesto de error tiene consecuencias escritas antes
- no se excluyen incidentes «que no cuentan»
- la ventana es de 30 días móviles
- existe potestad de frenar y se ha ejercido alguna vez
- el trabajo repetitivo tiene límite escrito
- hay criterios de entrada y de salida para aceptar servicios
- quien construye comparte la guardia
- gastar poco presupuesto también se revisa

Y el cierre que enlaza con la clase siguiente: la fiabilidad responde de que el sistema cumpla; queda quien responde de que un fallo no sea catastrófico ni deliberado. La ruta de seguridad es la materia de la clase 269.

Ejemplo trabajado

La ruta de fiabilidad en CloudShop: del panel que nadie miraba al presupuesto que paró un lanzamiento. Lo que sigue son los objetivos mal puestos que se corrigieron, la negociación que costó tres reuniones, y el servicio que hubo que devolver.

Punto de partida: los objetivos que no medían nada.

lo que había, 14 objetivos declarados por ingeniería

objetivo	valor	cumplido en 12 meses
disponibilidad de máquinas	99,95 %	12/12 meses
CPU media < 70 %	-	12/12
latencia media < 200 ms	-	11/12
errores 5xx < 0,1 %	99,9 %	12/12
...		

- 12 de 14 se cumplían todos los meses
- y en esos 12 meses hubo 34 incidentes de gravedad alta
- los objetivos no medían lo que fallaba

Y el ejemplo que lo dejó claro:

- el incidente del carrito perdido clase 261
 - el 9 % de los usuarios perdía la sesión
 - todas las peticiones devolvían 200
 - la latencia, normal
 - la CPU, normal
 - los 14 objetivos, cumplidos
- el usuario perdía su compra y ningún número lo reflejaba

La corrección: cuatro indicadores por recorrido.

se eligieron por RECORRIDO DE USUARIO, no por servicio

COMPRAR	
éxito	% de intentos de compra que se completan

latencia % que se completan en < 2 s
 BUSCAR Y VER CATÁLOGO
 éxito % de búsquedas con resultado
 latencia % en < 800 ms
 SEGUIMIENTO DE PEDIDO
 frescura % de consultas con estado de < 5 min de antigüedad
 DEVOLUCIONES
 éxito % de solicitudes que se registran

y se miden en el CLIENTE, no en el servidor
 → y ahí apareció una diferencia que nadie esperaba

Y la diferencia:

	el servidor	el cliente
éxito de compra	99,94 %	98,71 %

→ 1,23 puntos de diferencia
 → causas: plazos del cliente móvil, errores de red del usuario, y una versión de la aplicación con un fallo que reintentaba mal clase 201

→ y el 1,23 % eran unas 4.100 compras al mes
 → ninguna aparecía en ningún panel

La negociación del número: tres reuniones.

reunión 1 ingeniería propone 99,95 % para comprar
 negocio dice «¿por qué no 99,99?»

reunión 2 ingeniería lleva el coste
 99,9 % sistema actual + trabajo de fiabilidad ~0 adicional
 99,95 % + redundancia de la base entre zonas y vertido de carga +14.000 USD/mes
 99,99 % + segunda región activa, datos replicados, ensayos mensuales +112.000 USD/mes
 + 2 personas

y la pregunta que cambió la conversación
 «¿cuánto vale una hora de caída del flujo de compra?»
 → negocio lo calculó: ~31.000 USD

reunión 3 se acuerda
 comprar 99,95 %
 buscar 99,9 %
 seguimiento 99,5 %
 devoluciones 99,5 %

y el razonamiento escrito
 99,99 % costaría 1,34 M USD/año para evitar ~7,7 h de caída al año → ~239.000 USD de impacto
 → no se justifica

Y las consecuencias, acordadas por escrito:

presupuesto > 50 % despliegue libre; se pueden probar cosas
 presupuesto < 25 % solo cambios de bajo riesgo
 presupuesto agotado se congelan funciones nuevas del recorrido afectado hasta reponerlo

y la excepción
 producto puede pedir una revisión del objetivo, por escrito, con la nueva cifra y su vigencia

El trimestre en que se aplicó.

mes 2 el presupuesto de «comprar» cae al 11 %
 causa: dos incidentes y tres despliegues fallidos

mes 2 había un lanzamiento previsto para 9 días después

la conversación

no fue «¿lo paramos?»

fue «acordamos congelar; ¿queréis cambiar el acuerdo?»

lo que decidió negocio

el lanzamiento se retrasó 3 semanas

y el equipo dedicó ese tiempo a

la instancia con disco degradado clase 258

la verificación automática tras desplegar

y el vertido de carga clase 262

resultado

presupuesto del mes siguiente 94 %

y el lanzamiento salió sin incidentes

Y el dato que convenció a producto para el trimestre siguiente:

lanzamientos con presupuesto > 50 % 11

con incidente asociado 1

lanzamientos con presupuesto < 25 % 4

con incidente asociado 3

→ el presupuesto predecía el riesgo del lanzamiento

→ y a partir de ahí, producto empezó a mirarlo antes de planificar fechas

El servicio que hubo que devolver.

criterios de entrada, escritos

señales definidas y objetivos acordados

procedimientos en grado 2 o superior clase 259

despliegue progresivo con vuelta atrás clase 260

y guardia compartida con el equipo que construye

el servicio de recomendaciones entró en el mes 4

cumplía los cuatro

mes 9

interrupciones de guardia por ese servicio 41 %
del total

procedimientos actualizados por su equipo 0

incidentes con acciones cerradas 2/9

y su equipo había dejado de compartir la guardia

se aplicó el criterio de salida

→ se devolvió el servicio a su equipo, con 4 semanas de aviso

→ conversación muy incómoda

y lo que pasó a los 3 meses

interrupciones de ese servicio -74 %

procedimientos actualizados 6

y su equipo pidió volver a entrar

→ y volvió, cumpliendo los criterios

→ y el mecanismo funcionó: quien construye sufre lo que construye, y entonces lo arregla clase 111

Las cifras a los 18 meses.

objetivos definidos 14 6

que se cumplen siempre sin esfuerzo 12 0

medición en el cliente no sí

incidentes de gravedad alta 34/año 13/año

incidentes no reflejados en ningún objetivo 26/34 0/13

presupuesto agotado (trimestres) - 2 de 6

congelaciones aplicadas - 2

congelaciones ignoradas - 0

revisiones de objetivo pedidas por producto, por escrito - 1

trabajo repetitivo del equipo	71 %	38 %
servicios aceptados	-	17
servicios devueltos	-	1

La lección que esta clase deja: catorce objetivos se cumplían todos los meses mientras ocurrían 34 incidentes graves al año, porque medían máquinas y no recorridos. Y medir en el cliente en vez de en el servidor reveló 1,23 puntos de diferencia —unas 4.100 compras al mes— que ningún panel del servidor podía mostrar.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-22-specializations-certifications-career/268-ruta-site-reliability-engineer/lab.py
```

El laboratorio selecciona el motor de práctica decision y produce labresult.json. El escenario, sus comprobaciones y el artefacto esperado corresponden a esta clase; no requiere credenciales y deja explícito qué debe revalidarse en un sandbox real.

1. Lee exercise.steps y formula una predicción antes de ejecutar.
2. Ejecuta la práctica y verifica todos los elementos de checks.
3. Provoca el caso de negativetest y explica la señal observada.
4. Materializa sre-plan en evidence/ usando la plantilla indicada.
5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

El artefacto principal es una matriz de decisión ponderada y un ADR. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye sre-plan para el caso CloudShop. Incluye una alternativa descartada, un supuesto que pueda falsarse, una prueba de fallo y una decisión de rollback.

■ Criterio de aceptación

- [] lab.py termina con código 0 y genera JSON válido.
- [] La entrega conecta al menos tres requisitos con mecanismos verificables.
- [] Existe una prueba positiva y una prueba negativa con evidencia.
- [] Seguridad, costo y operación aparecen como decisiones, no como anexos.
- [] Se declara una limitación y una condición que obligaría a revisar el diseño.
- [] Otra persona puede repetir el recorrido sin conocimiento tácito.

■ Errores frecuentes

Síntoma	Causa probable	Corrección
Todos los objetivos se cumplen y hay incidentes graves cada mes	Los indicadores miden recursos y no recorridos de usuario	Define indicadores por recorrido, con umbral claro, y comprueba las dos direcciones: si empeora, ¿lo nota alguien? Si el usuario sufre, ¿esto lo refleja?
El presupuesto de error se informa y nunca cambia nada	No hay consecuencias acordadas antes ni potestad de aplicarlas	Escribe las consecuencias por tramo antes de necesitarlas y acuérdalas con producto; sin consecuencias es un panel bonito.
El equipo de fiabilidad hace las guardias de servicios que no construye	Se aceptaron servicios sin criterios de entrada y quien los construye dejó de recibir alertas	Fija criterios de entrada y de salida, exige guardia compartida y devuelve el servicio cuando dejen de cumplirse.
Se discute el objetivo cada vez que hay un lanzamiento	El número se declaró en vez de negociarse con su coste	Lleva el coste de cada nivel y el valor de una hora de caída; y deja que cambiar el acuerdo sea legítimo, pero explícito y escrito.
Se excluyen incidentes del cómputo porque fueron del proveedor	Se confunde responsabilidad con experiencia del usuario	El usuario no distingue de quién fue el fallo; si se excluye, el número deja de representarlo y pierde su función.
El presupuesto nunca se gasta y el equipo se siente bien	El objetivo está por debajo de lo que se consigue sin esfuerzo, o se está siendo conservador de más	Revisa también el presupuesto sin gastar: indica que se podría entregar más rápido o que el objetivo está mal puesto.

■ Seguridad, ética y costo

Trabaja con cuentas propias o sandboxes autorizados. No publiques secretos, identificadores reales ni datos personales. Antes de crear recursos pagos define presupuesto, etiquetas y comando de destrucción; después verifica que no queden recursos huérfanos. Los ejemplos locales enseñan contratos, pero no certifican cumplimiento ni disponibilidad de producción.

■ Preguntas de comprobación

1. ¿Qué tres cosas definen esta ruta frente a operar bien?
2. ¿Cómo se distingue un buen indicador de uno malo?
3. ¿Qué hace útil al presupuesto de error y qué lo convierte en decoración?
4. ¿Cómo empieza el modo de fracaso de la guardia prestada y cómo se evita?
5. ¿Cómo se negocia un objetivo con negocio sin preguntar qué disponibilidad quiere?

Referencias

- Beyer, B. y otros (2016). Site Reliability Engineering, caps. sobre SLO y presupuesto de error.
<<https://sre.google/sre-book/service-level-objectives/>>
- Google (2018). The Site Reliability Workbook, cap. «Implementing SLOs».
<<https://sre.google/workbook/implementing-slos/>>
- Wilkinson, A. y Hidalgo, A. (2020). Implementing Service Level Objectives.
<<https://www.oreilly.com/library/view/implementing-service-level/9781492076803/>>
- AWS (2024). Reliability Pillar: workload SLAs and SLOs.
<<https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/welcome.html>>
- Microsoft (2024). Azure Well-Architected Framework: reliability targets.
<<https://learn.microsoft.com/azure/well-architected/reliability/metrics>>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Anterior	Índice	Siguiente
← 267 · Ruta Platform Engineer	Parte 22 · Programa	269 · Ruta Cloud Security Engineer →

Evaluación — 268 Ruta Site Reliability Engineer

Preguntas

1. Explica el problema que resuelve esta capacidad sin mencionar un proveedor.
2. Identifica una frontera de responsabilidad y su propietario.
3. Compara dos alternativas mediante confiabilidad, seguridad, costo y operación.
4. ¿Qué demuestra el laboratorio y qué no puede demostrar?
5. Propón un caso límite o una prueba de fallo.

Reto verificable

Entrega sre-plan con diagrama o configuración, evidencia de ejecución, alternativa descartada, riesgo residual y criterio de rollback.

Criterio de aceptación

- ☐ Resultado reproducible por una segunda persona.
- ☐ Evidencia vinculada a cada afirmación técnica.
- ☐ Prueba negativa o de fallo incluida.
- ☐ Costos y permisos expresados con unidades y alcance.
- ☐ Limitaciones declaradas sin presentar la simulación como producción.

Escala

Nivel	Evidencia
A — competente	Cumple todo y defiende trade-offs con datos.
B — en desarrollo	Cumple lo esencial; falta profundidad en una dimensión.
C — inicial	Artefacto parcial o conclusión sin evidencia suficiente.
0	Sin entrega reproducible.

269 — Ruta Cloud Security Engineer

Parte: 22 — Especializaciones, certificaciones y práctica profesional Nivel: intermedio-avanzado · Horas estimadas: 4
Laboratorio: decision · Estado: EXECUTABLECORE

Propósito

La ruta de seguridad en la nube: conseguir que un fallo no sea catastrófico y que un ataque no sea rentable. La clase separa esta especialidad del cumplimiento normativo, da las competencias que se miden —modelado de amenazas, identidad, detección y respuesta—, y marca su modo de fracaso, que es el más caro de todos: convertirse en el equipo del no, al que se rodea.

Resultados de aprendizaje

Al finalizar podrás:

1. Distinguir seguridad de cumplimiento y saber para qué sirve cada uno.
2. Modelar amenazas de forma que produzca decisiones, no documentos.
3. Priorizar por explotabilidad e impacto, no por gravedad nominal.
4. Evitar el modo de fracaso del equipo al que se rodea.
5. Reconocer el techo de la ruta y qué la continúa.

Conceptos centrales

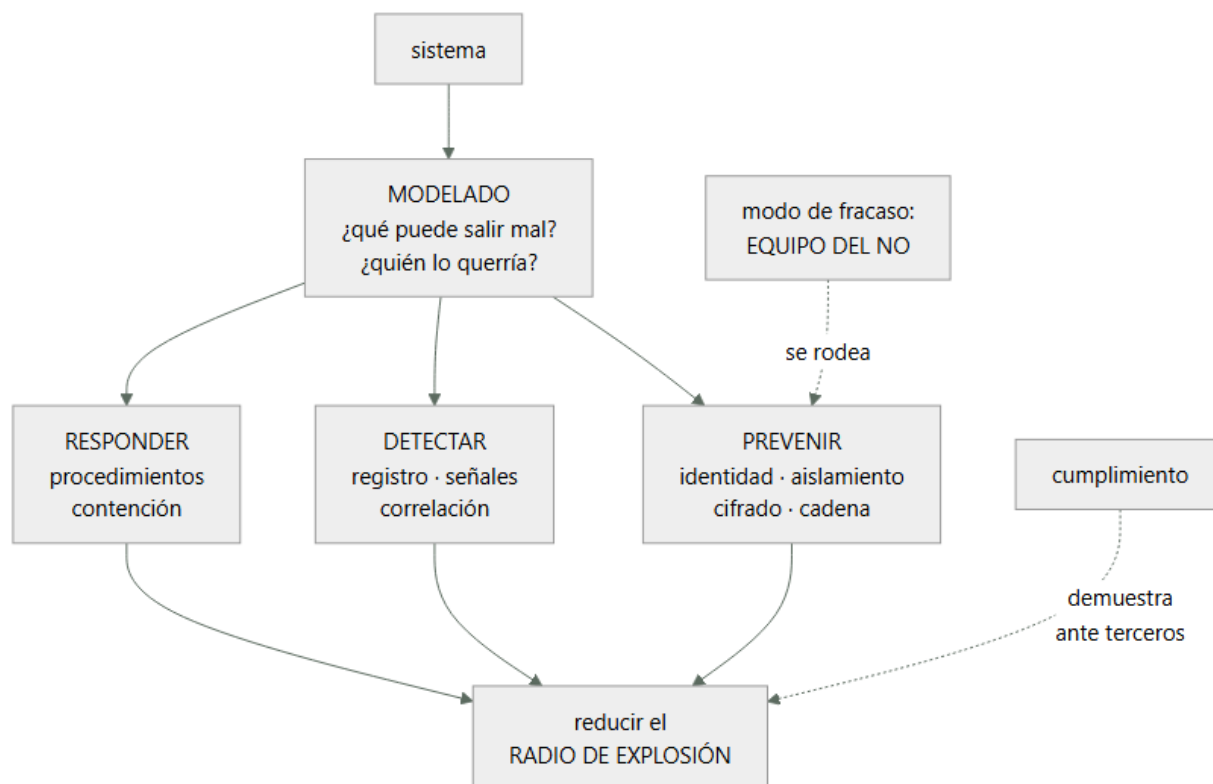
Concepto	Comprensión verificable
modelado de amenazas	Ejercicio de preguntar qué puede salir mal, quién lo querría y qué lo impediría. Produce decisiones, no documentos.
radio de explosión	Hasta dónde llega el daño si una parte cae. La variable que esta ruta reduce.
explotabilidad	Si la vulnerabilidad es alcanzable y aprovechable en este sistema concreto. Domina sobre la gravedad nominal.
camino pavimentado seguro	Hacer que lo seguro sea lo fácil. La alternativa a la norma que hay que recordar.
cumplimiento	Demostrar ante un tercero que se hace lo que se dice. Necesario, y distinto de estar seguro.
equipo del no	Modo de fracaso en que seguridad bloquea sin alternativa y los equipos aprenden a esquivarla.

Modelo mental

Una especialización combina fundamentos, evidencia de proyectos y juicio bajo restricciones; una insignia sin práctica no sustituye esa combinación.

Aplicado a esta clase, separa siempre cuatro planos: intención (qué necesita el usuario), configuración (qué declaramos), estado observado (qué existe de verdad) y evidencia (cómo sabemos que cumple). Confundirlos produce diseños que se ven correctos en un diagrama pero fallan al operar.

■ Flujo de razonamiento



Desarrollo

1. Seguridad no es cumplimiento

Los dos son necesarios y se confunden constantemente, con consecuencias caras en las dos direcciones.

SEGURIDAD

reducir la probabilidad y el impacto de que alguien
haga algo que no debería
→ se mide por lo que pasa cuando se intenta

CUMPLIMIENTO

demostrar ante un tercero que se hace lo que se dice
→ se mide por evidencia auditable

- un sistema puede cumplir y ser inseguro
- y puede ser seguro y no poder demostrarlo
- y el error caro es dejar que el calendario de auditoría
fije las prioridades de seguridad

Y lo que esta ruta hace de verdad, en tres verbos:

PREVENIR

identidad y privilegio mínimo	clase 231
aislamiento de red y de cuentas	clases 219, 231
cifrado y gestión de claves	clase 197
cadena de suministro: firma y procedencia	clase 216
y valores por defecto seguros	ley 26

DETECTAR

registro completo e inalterable
señales de comportamiento anómalo
y tiempo hasta detectar, medido

RESPONDER

contener, erradicar, recuperar	
con procedimientos ejecutables	clase 259
y ensayados	clase 261

- y la mayoría de los equipos gastan casi todo en
prevenir

→ y la diferencia entre un incidente y una catástrofe la hacen los otros dos

Y la variable que resume la ruta:

EL RADIO DE EXPLOSIÓN

«si esta credencial se filtra, ¿hasta dónde llega?»
«si este servicio se compromete, ¿qué puede tocar?»
«si esta cuenta cae, ¿qué otras caen con ella?»

→ y casi todo el trabajo técnico de la ruta es reducirlo

cuentas separadas por entorno	clase 219
permisos por propósito	clase 231
credenciales temporales	clase 256
copias que sobreviven al administrador	clase 255
y segmentación de red	clase 194

→ porque asumir que nada fallará no es una estrategia

2. Modelado de amenazas que produce decisiones

El modelado de amenazas tiene mala fama porque suele producir documentos. Hecho bien, produce cambios concretos.

LAS CUATRO PREGUNTAS

- 1 ¿qué estamos construyendo?
→ un diagrama con los límites de confianza
→ y lo que no está en el diagrama no se analiza
ley 24
- 2 ¿qué puede salir mal?
→ por cada flujo que cruza un límite
- 3 ¿qué vamos a hacer?
→ mitigar, aceptar, transferir o eliminar
- 4 ¿lo hicimos bien?
→ y esta cuarta es la que casi siempre falta

→ una sesión de 90 minutos con las personas que lo construyen

→ y sale con acciones, dueño y plazo, o no sirvió

Y dónde mirar, por rendimiento:

LOS LÍMITES DE CONFIANZA, que es donde está casi todo

entrada de usuario a sistema	
servicio a servicio	clase 231
sistema a proveedor externo	clase 188
plano de control a plano de datos	
y persona a sistema	clase 256

Y LAS PREGUNTAS QUE MÁS DESCUBREN

«¿qué pasa si este componente miente?»
«¿qué puede hacer esta identidad que no necesita?»
«¿quién puede borrar las copias?» clase 255
«¿quién puede desactivar el registro?»
«¿qué credencial hay aquí que no caduca?»
y «si esto se compromete, ¿cómo nos enteraríamos?»

→ la última suele no tener respuesta

→ y esa ausencia es el hallazgo más valioso de la sesión

Y el criterio de prioridad, que corrige el error más común:

NO SE PRIORIZA POR GRAVEDAD NOMINAL

una vulnerabilidad «crítica» en una biblioteca que el código nunca invoca, sin exposición externa, es menos urgente que una «media» en el borde

SE PRIORIZA POR

¿es alcanzable desde fuera?
¿está en el camino de ejecución?
¿hay explotación conocida en circulación?
¿qué radio de explosión tiene si se aprovecha?
¿y qué compensaciones existen ya?

→ y esto es lo que convierte 3.000 hallazgos en 40

acciones clase 216
→ y una lista de 3.000 no se arregla: se ignora

3. El modo de fracaso: el equipo del no

Es el modo de fracaso más caro de todas las rutas, porque su resultado no es lentitud: es que la seguridad deja de existir donde importa.

CÓMO EMPIEZA

seguridad bloquea algo con razón
→ y sin ofrecer alternativa
el equipo necesita entregar
→ y encuentra otro camino

QUÉ PRODUCE

el trabajo se hace igual, fuera del alcance de seguridad
→ cuentas personales, servicios sin registrar,
credenciales en repositorios privados
seguridad deja de saber qué existe
→ y pierde la única cosa que necesitaba: visibilidad
y se entera de los sistemas cuando fallan

→ es la ley 16 en su forma más costosa
→ y el daño no es que se salte una norma: es que el
sistema real se vuelve invisible

Y cómo se sale, que define el nivel 4 de esta ruta:

- 1 NUNCA BLOQUEAR SIN ALTERNATIVA
«no puedes usar eso» → «usa esto, que ya está listo»
→ y si no hay alternativa, el trabajo de seguridad es
construirla, no prohibir
- 2 CAMINO PAVIMENTADO SEGURO
la plantilla ya trae permisos mínimos, cifrado,
registro y secretos gestionados clase 267
→ y entonces lo seguro es lo fácil
→ y la norma escrita deja de ser necesaria
- 3 POLÍTICAS EN MODO AVISO ANTES DE BLOQUEAR clase 217
→ se mide el impacto antes de imponerlo
- 4 Y AMNISTÍA PARA LO QUE APAREZCA
«cuéntanos qué tienes fuera y te ayudamos a traerlo,
sin consecuencias»
→ recupera visibilidad, que es lo que se había perdido

Y el segundo modo de fracaso, el del teatro:

SEGURIDAD DE INFORME

paneles llenos, informes mensuales, cientos de reglas
→ y ningún ensayo de respuesta clase 261
→ y nadie ha comprobado si la detección funciona

la prueba

haz algo que DEBERÍA detectarse y cronometra
→ crear una clave de acceso permanente
→ dar permisos amplios a un rol
→ exfiltrar un fichero grande a un destino externo

→ y si nadie lo detecta, las reglas son decoración
→ este ejercicio, hecho una vez, reordena las
prioridades del año

4. Niveles, evidencia y techo

Lo que se mide en esta ruta, por nivel.

NIVEL 2 · RESUELVO

configura identidad, red y cifrado correctamente
revisa permisos y encuentra los excesivos
gestiona secretos y su rotación clase 197
responde a un incidente siguiendo el procedimiento
y prioriza vulnerabilidades por explotabilidad

NIVEL 3 · DISEÑO

modela amenazas y produce decisiones
diseña el aislamiento y el radio de explosión
monta detección que se ha comprobado que detecta
dirige un incidente de seguridad, con su comunicación
y negocia riesgo aceptado, por escrito y con dueño

NIVEL 4 · CAMBIO EL SISTEMA

lo seguro es lo fácil, y la adopción se mide
los equipos llaman a seguridad ANTES de construir
→ y esa es la señal definitiva de que la ruta funciona
y el riesgo aceptado es una decisión de negocio
registrada, no un silencio

Y la evidencia que vale:

LO QUE NO VALE

«implantamos la herramienta X»
«pasamos la auditoría»
→ describe actividad, no efecto

LO QUE VALE

«el tiempo hasta detectar una credencial expuesta pasó
de 9 días a 40 minutos, y así lo probamos»
«eliminamos 41 claves permanentes y el acceso pasó a
sesiones temporales auditadas»
«de 3.000 hallazgos, 40 eran alcanzables; los cerramos
en 6 semanas y el resto se documentó con su motivo»
«un servicio comprometido ya no puede tocar producción
porque están en cuentas distintas»

→ efecto, mecanismo y cifra clase 275

Y el techo:

EL TECHO

los controles funcionan, la detección detecta y los
equipos consultan antes de construir
→ y lo que limita entonces es el riesgo que la
organización acepta, que es una decisión de
dirección

continuaciones

- a ARQUITECTURA clase 272
si el límite es cómo está construido el sistema
- b GOBIERNO Y RIESGO
si el límite es cómo se decide qué se acepta
- c o dirección de seguridad
donde el trabajo es presupuesto, personas y
prioridades

Y la lista de comprobación de la clase:

- distingo seguridad de cumplimiento y no dejo que la auditoría fije las prioridades
- hay inversión en detectar y responder, no solo en prevenir
- sé el radio de explosión de mis credenciales y cuentas
- el modelado de amenazas sale con acciones y dueño
- sé responder a «si esto se compromete, ¿cómo nos enteraríamos?»
- priorizo por explotabilidad, no por gravedad nominal
- nunca bloqueo sin ofrecer alternativa
- existe camino pavimentado seguro y mido su adopción
- las políticas pasan por modo aviso antes de bloquear
- he comprobado que la detección detecta, cronometrando
- el riesgo aceptado está escrito, con dueño y vigencia
- los equipos me llaman antes de construir, no después

Y el cierre que enlaza con la clase siguiente: seguridad responde de que el fallo no sea catastrófico; queda quien responde de que el gasto sea una decisión y no una consecuencia. La ruta de gestión económica de la nube es la materia de la clase 270.

Ejemplo trabajado

La función de seguridad de CloudShop, dos años. Lo que sigue es la prueba que descubrió que la detección no detectaba, la lista de 3.100 hallazgos convertida en 47 acciones, y la amnistía que reveló lo que se había construido fuera.

Prueba 1 · ¿Detecta la detección?

ejercicio anunciado a dirección, no al equipo de guardia
cinco acciones que deberían detectarse, cronometradas

acción	detectada	tiempo
crear una clave de acceso permanente	no	-
dar permiso administrativo a un rol de servicio	sí	4 min
desactivar el registro de auditoría en una cuenta	no	-
copiar 2 GB a un destino externo	no	-
acceder a la base de producción desde una IP nueva	sí	31 min
detectadas	2 de 5	

Y lo que se descubrió al investigar por qué:

la regla de «clave permanente creada» existía
→ enviaba a un canal retirado hacía 7 meses
ley 15
la de «registro desactivado» existía
→ y solo cubría 2 de las 9 cuentas
clase 219
la de exfiltración
→ nunca se había escrito; se daba por cubierta por otra
→ tres de cinco fallos eran de configuración, no de capacidad
→ el ejercicio costó 4 horas y reordenó el plan del año

Y lo que se hizo:

cobertura de reglas verificada por cuenta y por región
cada regla con una PRUEBA que la dispara, ejecutada mensualmente
clase 261
→ y si la prueba no dispara la alerta, la regla está rota
reejecución del ejercicio a los 5 meses
detectadas 5 de 5
tiempo máximo 6 minutos

Prueba 2 · De 3.100 hallazgos a 47 acciones.

el escáner producía	3.100
críticos	287
altos	941

y llevaba 14 meses sin que nadie lo mirara
→ porque una lista de 3.100 no se arregla: se ignora

Y el filtro que se aplicó, por orden:

3.100 todos los hallazgos
1.412 en componentes que llegan a producción
610 en el camino de ejecución del código
188 alcanzables desde fuera o desde entrada de usuario
94 sin compensación existente
47 con explotación conocida en circulación o con radio de explosión alto
→ 47 acciones, cerradas en 6 semanas
→ y de los 287 «críticos» originales, 19 quedaron en la lista final
→ y 3 de los 47 eran de gravedad «media»

Y lo que se hizo con los 3.053 restantes:

se documentaron por lote, con el motivo
«no alcanzable: componente no desplegado»
«no en camino de ejecución»
«compensado por segmentación de red»

y se automatizó el filtro
→ el informe semanal pasó de 3.100 líneas a entre 4 y
12 acciones
→ y esas sí se atendían

tiempo medio de cierre de un hallazgo alcanzable
antes no medible (nadie miraba)
después 9 días

Prueba 3 · La amnistía.

situación
seguridad había bloqueado, en 18 meses
uso de una base gestionada concreta
despliegue directo desde portátiles
y tres servicios de terceros
sin ofrecer alternativa en ninguno de los tres casos

la amnistía
«cuéntanos qué tienes funcionando fuera del alcance;
te ayudamos a traerlo; sin consecuencias; 4 semanas»

lo que apareció

cuentas de nube personales con carga real	6
servicios de terceros contratados con tarjeta de empresa	11
repositorios privados con credenciales	4
una base de datos con datos de clientes en una cuenta no inventariada	1
y un servicio expuesto a internet sin cortafuegos	1

Y la lectura:

los seis primeros casos existían porque la alternativa oficial tardaba semanas
→ y el bloqueo no impidió el trabajo: lo hizo invisible

y el caso de la base con datos de clientes
→ llevaba 11 meses
→ sin copias, sin registro, sin cifrado en reposo
→ y era exactamente el riesgo que la norma bloqueada pretendía evitar

→ el bloqueo produjo el riesgo que quería evitar
→ ley 16, en su versión más cara

Y lo que se cambió:

camino pavimentado seguro clase 267
plantilla con permisos mínimos, cifrado, registro y
secretos gestionados, lista para usar
tiempo desde la petición hasta tener entorno
3 semanas → 4 horas

y la regla de trabajo del equipo
«no se bloquea nada sin alternativa disponible el mismo día»

resultado a los 12 meses

recursos fuera del inventario detectados	1.107 → 12
peticiones de excepción	34 → 6
y equipos que consultan a seguridad ANTES de construir	2 → 11

Las cifras a los dos años.

DETECCIÓN		
acciones de prueba detectadas	2 de 5	5 de 5
tiempo máximo hasta detectar	n/d	6 min

reglas con prueba de disparo mensual	0	41
cobertura de registro por cuenta	2 de 9	9 de 9
PREVENCIÓN		
claves de acceso permanentes	41	0
roles con permisos administrativos	23	4
cuentas con separación por entorno	no	sí
copias que sobreviven al administrador	no	sí
GESTIÓN		
hallazgos en el informe semanal	3.100	4-12
tiempo de cierre de hallazgo alcanzable	n/d	9 días
riesgos aceptados con dueño y vigencia	0	14
RELACIÓN CON LOS EQUIPOS		
recursos fuera del inventario	1.107	12
equipos que consultan antes de construir	2	11
peticiones de excepción	34	6

Y el dato que la persona responsable puso primero:

equipos que consultan a seguridad ANTES de construir
2 → 11 de 14

→ y esa es la única métrica que no se puede falsear
→ nadie consulta voluntariamente al equipo del no

La lección que esta clase deja: la detección estaba montada y detectaba 2 de 5 acciones que debía detectar, y tres de los tres fallos eran de configuración —un canal retirado, dos cuentas de nueve cubiertas, una regla que nunca se escribió—, no de capacidad. Y el bloqueo sin alternativa produjo exactamente el riesgo que pretendía evitar: una base con datos de clientes, once meses sin copias, sin registro y sin cifrado, en una cuenta que nadie sabía que existía.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-22-specializations-certifications-career/269-ruta-cloud-security-engineer/lab.py
```

El laboratorio selecciona el motor de práctica decision y produce labresult.json. El escenario, sus comprobaciones y el artefacto esperado corresponden a esta clase; no requiere credenciales y deja explícito qué debe revalidarse en un sandbox real.

1. Lee exercise.steps y formula una predicción antes de ejecutar.
2. Ejecuta la práctica y verifica todos los elementos de checks.
3. Provoca el caso de negativetest y explica la señal observada.
4. Materializa security-engineer-plan en evidence/ usando la plantilla indicada.
5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

El artefacto principal es una matriz de decisión ponderada y un ADR. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye security-engineer-plan para el caso CloudShop. Incluye una alternativa descartada, un supuesto que pueda falsarse, una prueba de fallo y una decisión de rollback.

■ Criterio de aceptación

- [] lab.py termina con código 0 y genera JSON válido.
- [] La entrega conecta al menos tres requisitos con mecanismos verificables.
- [] Existe una prueba positiva y una prueba negativa con evidencia.
- [] Seguridad, costo y operación aparecen como decisiones, no como anexos.

- [] Se declara una limitación y una condición que obligaría a revisar el diseño.
- [] Otra persona puede repetir el recorrido sin conocimiento tácito.

■ Errores frecuentes

Síntoma	Causa probable	Corrección
Se pasa la auditoría y aun así ocurre una brecha	El calendario de cumplimiento fijó las prioridades de seguridad	Separa las dos funciones: cumplimiento demuestra ante terceros, seguridad reduce probabilidad e impacto; prioriza por radio de explosión.
Hay miles de hallazgos y no se arregla ninguno	Se prioriza por gravedad nominal en vez de por explotabilidad	Filtra por alcanzable, en camino de ejecución, con explotación conocida y radio de explosión; documenta el resto por lote con su motivo.
Aparecen sistemas que nadie sabía que existían	Se bloqueó sin ofrecer alternativa y el trabajo se hizo fuera del alcance	No bloquee sin alternativa disponible, monta el camino pavimentado seguro y ofrece amnistía para recuperar visibilidad.
Las reglas de detección existen y no detectan	Nadie ha comprobado si disparan: canales retirados, cobertura parcial, reglas que nunca se escribieron	Ejecuta acciones que deberían detectarse y cronometra; cada regla necesita una prueba de disparo periódica.
El modelado de amenazas produce documentos y ningún cambio	Falta la cuarta pregunta y no salen acciones con dueño y plazo	Cierra la sesión con acciones asignadas y revisa después si se hicieron; y pregunta siempre cómo os enteraréis si eso se comprometiera.
Seguridad se entera de los proyectos cuando ya están en producción	Consultar es caro y bloquea, así que se evita	Mide cuántos equipos consultan antes de construir; si no crece, el problema es tu proceso, no su disciplina.

■ Seguridad, ética y costo

Trabaja con cuentas propias o sandboxes autorizados. No publiques secretos, identificadores reales ni datos personales. Antes de crear recursos pagos define presupuesto, etiquetas y comando de destrucción; después verifica que no queden recursos huérfanos. Los ejemplos locales enseñan contratos, pero no certifican cumplimiento ni disponibilidad de producción.

■ Preguntas de comprobación

1. ¿En qué se diferencian seguridad y cumplimiento y qué error produce confundirlos?
2. ¿Qué es el radio de explosión y qué medidas lo reducen?
3. ¿Qué criterio convierte miles de hallazgos en unas pocas acciones?
4. ¿Cómo empieza el modo de fracaso del equipo del no y qué produce?
5. ¿Cómo se comprueba que la detección detecta?

Referencias

- Shostack, A. (2014). Threat Modeling: designing for security. <<https://shostack.org/books/threat-modeling-book>>
- NIST (2018). Cybersecurity Framework — identificar, proteger, detectar, responder, recuperar. <<https://www.nist.gov/cyberframework>>
- AWS (2024). Security Pillar, Well-Architected Framework. <<https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/welcome.html>>
- Microsoft (2024). Azure security best practices and patterns. <<https://learn.microsoft.com/azure/security/fundamentals/best-practices-and-patterns>>
- Google Cloud (2024). Security foundations blueprint. <<https://cloud.google.com/architecture/security-foundations>>

- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 22 en PDF · Manual integral

Anterior	Índice	Siguiente
← 268 · Ruta Site Reliability Engineer	Parte 22 · Programa	270 · Ruta FinOps Practitioner →

Evaluación — 269 Ruta Cloud Security Engineer

Preguntas

1. Explica el problema que resuelve esta capacidad sin mencionar un proveedor.
2. Identifica una frontera de responsabilidad y su propietario.
3. Compara dos alternativas mediante confiabilidad, seguridad, costo y operación.
4. ¿Qué demuestra el laboratorio y qué no puede demostrar?
5. Propón un caso límite o una prueba de fallo.

Reto verificable

Entrega security-engineer-plan con diagrama o configuración, evidencia de ejecución, alternativa descartada, riesgo residual y criterio de rollback.

Criterio de aceptación

- ☐ Resultado reproducible por una segunda persona.
- ☐ Evidencia vinculada a cada afirmación técnica.
- ☐ Prueba negativa o de fallo incluida.
- ☐ Costos y permisos expresados con unidades y alcance.
- ☐ Limitaciones declaradas sin presentar la simulación como producción.

Escala

Nivel	Evidencia
A — competente	Cumple todo y defiende trade-offs con datos.
B — en desarrollo	Cumple lo esencial; falta profundidad en una dimensión.
C — inicial	Artefacto parcial o conclusión sin evidencia suficiente.
0	Sin entrega reproducible.

270 — Ruta FinOps Practitioner

Parte: 22 — Especializaciones, certificaciones y práctica profesional Nivel: intermedio-avanzado · Horas estimadas: 4
Laboratorio: decision · Estado: EXECUTABLECORE

Propósito

La ruta económica de la nube: conseguir que el gasto sea una decisión y no una consecuencia. La clase separa esta especialidad de «recortar factura», da el modelo que la sostiene —visibilidad, atribución, unidad económica y decisión al diseñar—, y marca su modo de fracaso: medir el ahorro conseguido en vez del coste evitado, que es donde está casi todo el valor.

Resultados de aprendizaje

Al finalizar podrás:

1. Distinguir reducción de factura de gestión económica del sistema.
2. Atribuir el coste a quien lo decide, con la granularidad justa.
3. Definir la unidad económica que hace comparable el gasto.
4. Intervenir donde el coste se decide: el diseño, no la factura.
5. Evitar medirse por el ahorro y perder el coste evitado.

Conceptos centrales

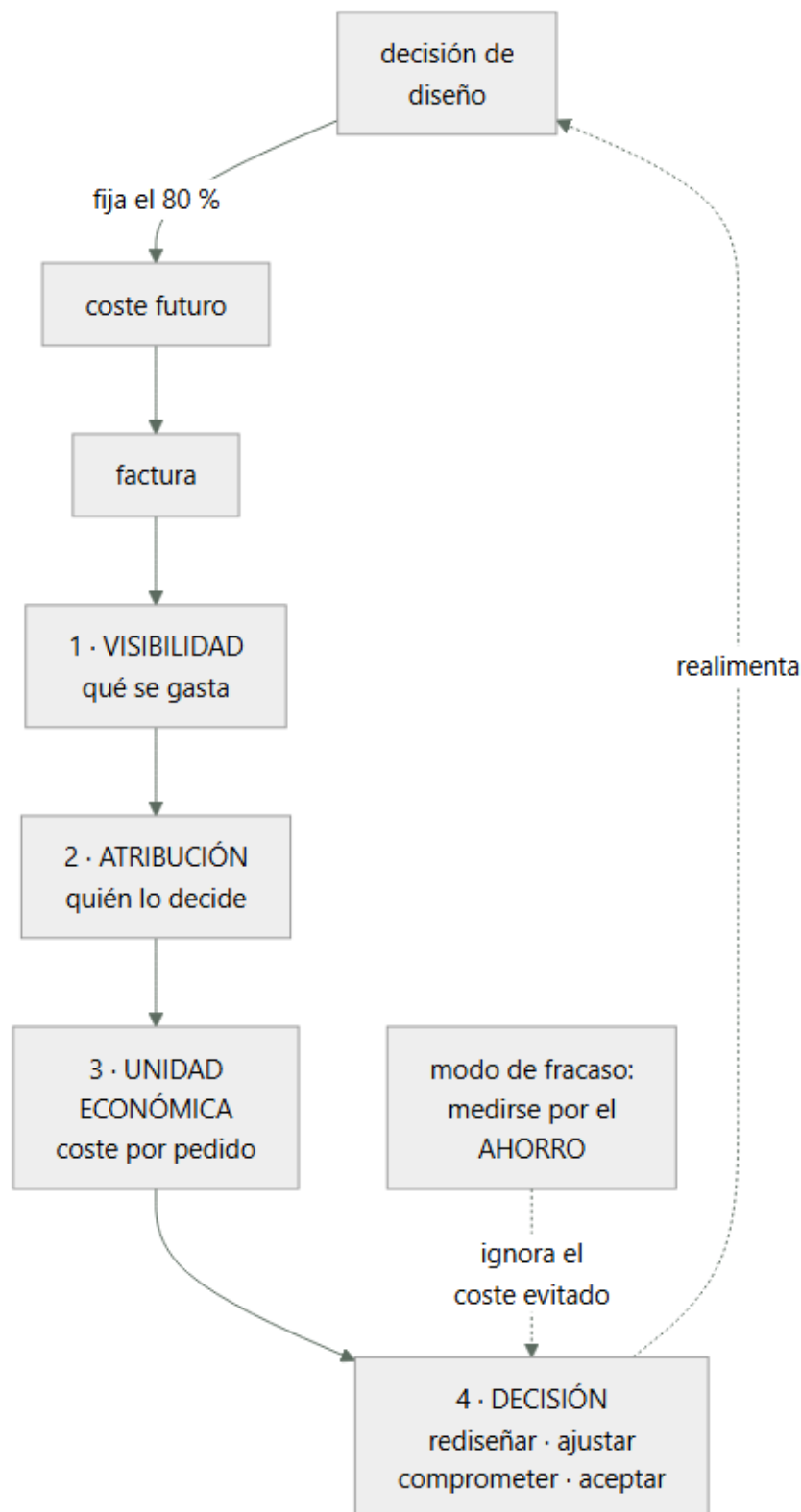
Concepto	Comprensión verificable
unidad económica	Coste por unidad de negocio: por pedido, por cliente, por consulta. Hace comparable el gasto en el tiempo.
atribución	Asignar cada coste a quien puede decidir sobre él. Sin ella no hay decisión posible.
coste evitado	El gasto que no llegó a existir porque la decisión de diseño fue otra. Invisible en la factura.
compromiso	Descuento a cambio de uso garantizado. Palanca financiera, no de ingeniería.
arquitectura del coste	Las decisiones de diseño que fijan la mayor parte del gasto antes de facturar nada.
coste no atribuido	Lo que no tiene dueño. Crece siempre, porque nadie decide sobre ello.

Modelo mental

Una especialización combina fundamentos, evidencia de proyectos y juicio bajo restricciones; una insignia sin práctica no sustituye esa combinación.

Aplicado a esta clase, separa siempre cuatro planos: intención (qué necesita el usuario), configuración (qué declaramos), estado observado (qué existe de verdad) y evidencia (cómo sabemos que cumple). Confundirlos produce diseños que se ven correctos en un diagrama pero fallan al operar.

■ Flujo de razonamiento



Desarrollo

1. No es recortar la factura

La especialidad se confunde con una campaña de recortes. Y una campaña de recortes tiene un final; esta ruta no.

RECORTAR FACTURA

apagar lo que no se usa, redimensionar, comprar compromisos

- efecto real, y acotado
- se agota: hay un suelo

GESTIÓN ECONÓMICA DEL SISTEMA

que cada decisión de diseño se tome sabiendo lo que cuesta

- y eso no se agota, porque el sistema no deja de cambiar

- y la diferencia se ve en una cifra: el coste por unidad de negocio
- una empresa que crece puede tener factura creciente y coste por pedido decreciente
- y esa es la señal de que la ruta funciona

Y la razón por la que la factura es un mal punto de intervención:

EL COSTE SE DECIDE AL CREAR, NO AL PAGAR	ley 14
el tipo de base de datos	
el formato y la partición de los datos	clase 243
cuántas veces se llama a un servicio	clase 106
si se replica entre regiones	clase 187
cuánto se registra y cuánto se conserva	clase 211
y si algo consulta o escucha	clase 210

- cuando llega la factura, esas decisiones ya están tomadas
- cambiar una decisión de diseño cuesta semanas; ajustar un tamaño cuesta minutos y ahorra mucho menos

Y las cuatro etapas del trabajo, en orden:

1 VISIBILIDAD

qué se gasta, en qué, con qué granularidad y con qué retraso

- y sin esto no hay conversación posible

2 ATRIBUCIÓN

cada coste a quien puede decidir sobre él

- por etiqueta, por cuenta, por proyecto
clases 219, 220, 253
→ y lo no atribuido crece siempre ley 20

3 UNIDAD ECONÓMICA

coste por pedido, por cliente activo, por consulta
→ convierte «gastamos 340.000» en «cuesta 0,41 por
pedido»

- y eso sí se puede juzgar y comparar

4 DECISIÓN

rediseñar, ajustar, comprometer o aceptar

- con dueño y con plazo

2. Atribución y unidad económica

Las dos piezas donde esta ruta gana o pierde credibilidad.

LA ATRIBUCIÓN

la pregunta no es «¿de qué equipo es este recurso?»
es «¿quién puede tomar una decisión que cambie este
coste?»

y hay costes que no se atribuyen bien
recursos compartidos: red, registros, plataforma
→ repartir por uso si se puede medir

- si no, por una regla ACORDADA, no perfecta
- una regla acordada e imperfecta funciona; ninguna regla no

y el objetivo realista

- er objetivo: calista
> 95 % del coste atribuido
y lo no atribuido, decreciente y con dueño del
problema

Y el error caro de la atribución:

ATRIBUIR SIN DAR PALANCA

- «tu equipo gasta 41.000 al mes»
- y el equipo no puede cambiar la instancia porque la fija la plataforma
- y no puede reducir el tráfico porque lo genera otro

- atribuir sin capacidad de decidir produce frustración, no ahorro
- la atribución tiene que ir con la palanca

Y la unidad económica, que es lo que hace la conversación posible:

CÓMO SE ELIGE

- algo que el negocio ya cuenta
- pedidos, clientes activos, consultas atendidas,
- minutos servidos, documentos procesados
- y que crece con el uso real

QUÉ REVELA

- el coste absoluto sube y la unidad baja → sano
- ambos suben → el sistema escala mal
- el absoluto baja y la unidad sube → se recortó algo que importaba

- y esta tabla convierte una discusión de presupuesto en una de ingeniería

Y las palancas, en orden de rendimiento:

- 1 ELIMINAR TRABAJO QUE NO HACE FALTA
 - sondeos que podrían ser eventos clase 210
 - reintentos excesivos clase 201
 - datos que se copian tres veces
 - registros que nadie consulta y se guardan 400 días
 - entornos de ensayo encendidos de noche
 - aquí suele estar el mayor rendimiento y nadie lo mira
- 2 CAMBIAR LA FORMA DE LOS DATOS
 - formato columnar, partición, compresión clase 243
 - factores de 5 a 50 en analítica
- 3 MOVER TRABAJO A DONDE ES BARATO
 - lotes en horas valle, capacidad interrumpible
 - caché en el borde clase 207
- 4 AJUSTAR TAMAÑOS
 - efecto real y limitado
- 5 COMPROMISOS Y DESCUENTOS
 - palanca financiera; se aplica AL FINAL
 - comprometerse sobre una arquitectura mala congela la arquitectura mala

3. El modo de fracaso: medirse por el ahorro

Es el modo de fracaso más sutil, porque la métrica equivocada parece la correcta.

SI LA RUTA SE MIDE POR «AHORRO CONSEGUIDO»

- el incentivo es dejar que el gasto crezca para poder recortarlo
- el trabajo de prevención no cuenta, porque no genera ahorro
- y cuando ya no queda nada obvio que recortar, la función parece agotada

- y la mayor parte del valor está justo en lo que esa métrica no ve

Y qué es el coste evitado y cómo se cuenta:

COSTE EVITADO

- el diseño se cambió antes de construir y el gasto nunca existió

ejemplos medibles

- se eligió partición por fecha en vez de exploración completa

- el mismo informe cuesta 12 USD en vez de 340

- se sustituyó un sondeo por eventos antes de lanzar

- 22 % menos peticiones, que nunca se facturaron

- se descartó la réplica activa entre regiones porque

- el objetivo de recuperación no la necesitaba

- 112.000 USD/mes que no se gastaron

cómo se registra

- en el momento de la decisión, con la alternativa

- descartada y su coste estimado

- y firmado por quien decide

- si se apunta después, nadie se lo cree

Y las métricas correctas de la ruta:

- coste por unidad de negocio, y su tendencia

- % del coste atribuido

- % de decisiones de diseño con coste estimado ANTES

- coste evitado registrado, con la alternativa

- precisión de la previsión mensual

- y esta última importa más de lo que parece: sin

- previsión fiable, finanzas no confía en el resto

- y recursos sin dueño y sin uso

- ninguna de estas es «ahorro conseguido»

- y el ahorro aparece igual, como consecuencia

Y el segundo modo de fracaso, el del panel:

PANELES DE COSTE QUE NADIE MIRA

- se monta la visibilidad y ahí se queda

- porque ver el coste no cambia nada si no llega a quien decide, cuando decide

LO QUE SÍ FUNCIONA

- el coste estimado en la propuesta de cambio

- antes de aprobar, no en el informe mensual

- alertas de anomalía por servicio, con dueño

- y una revisión periódica corta, con acciones

- el coste tiene que aparecer donde se toma la decisión

- que es exactamente el mismo principio del camino

- pavimentado

clases 267, 269

4. Niveles, evidencia y techo

Qué se espera por nivel en esta ruta.

NIVEL 2 · RESUELVO

- monta visibilidad y atribución con etiquetas

- encuentra desperdicio y lo elimina

- entiende el modelo de precios de lo que usa

- y explica una factura línea a línea

NIVEL 3 · DISEÑO

- define la unidad económica con negocio

- estima el coste de una arquitectura ANTES de construirla

- negocia compromisos con datos de uso real

- decide entre alternativas con coste, riesgo y

- rendimiento juntos

- y dice que una optimización no compensa

NIVEL 4 · CAMBIO EL SISTEMA

- el coste aparece en el momento de la decisión, siempre

- los equipos estiman su coste sin que se lo pidan

- y la conversación con negocio es sobre unidad económica,

- no sobre factura

Y la evidencia que vale:

LO QUE NO VALE

- «ahorramos un 30 %»

→ sin decir de qué base ni si el sistema creció

LO QUE VALE

«el coste por pedido pasó de 0,41 a 0,17 mientras los pedidos crecían un 34 %»
«el 96 % del coste está atribuido y lo no atribuido baja cada mes»
«registramos 340.000 USD/año de coste evitado, con la alternativa y quien la descartó»
«la previsión mensual falla menos del 4 %»

→ efecto, mecanismo y cifra clase 275

Y el techo:

EL TECHO

el coste está atribuido, la unidad baja y las decisiones llevan cifra
→ y lo que limita entonces es la arquitectura o la estrategia de producto

continuaciones

- a ARQUITECTURA clase 272
si el gasto lo fija cómo está construido
- b producto o dirección financiera de tecnología
si la conversación pasa a ser de márgenes y precios
- c o plataforma clase 267
si lo que falta es que hacer lo barato sea lo fácil

Y la lista de comprobación de la clase:

- distingo recortar factura de gestionar el coste
- tengo unidad económica definida con negocio
- miro su tendencia, no el coste absoluto
- más del 95 % del coste está atribuido
- quien recibe la atribución tiene palanca para cambiarla
- el coste aparece en la propuesta de cambio, antes de aprobar
- registro coste evitado en el momento de decidir
- no me mido por ahorro conseguido
- ataco primero el trabajo innecesario, no los tamaños
- los compromisos van al final, sobre arquitectura estable
- hay alertas de anomalía por servicio, con dueño
- la previsión mensual es fiable

Y el cierre que enlaza con la clase siguiente: quedan las dos rutas que trabajan sobre el material del que salen las decisiones. Datos e inteligencia artificial, con lo que la parte 20 dejó demostrado, es la materia de la clase 271.

Ejemplo trabajado

La función económica de CloudShop, dos años. Lo que sigue es la atribución que empezó en el 38 %, la unidad económica que cambió la conversación con dirección, y el registro de coste evitado que salvó la función cuando el ahorro obvio se agotó.

Año 0 · El punto de partida.

factura mensual	341.000 USD
crecimiento interanual	+41 %
coste atribuido a un equipo	38 %

y la conversación con dirección, cada trimestre
«la nube sube mucho»
«es que crecemos»
→ sin datos que resolvieran la discusión

Y el desglose de lo no atribuido, que fue el primer trabajo:

recursos sin etiqueta	27 %	44 %
servicios compartidos (red, registros)	19 %	31 %
cuentas antiguas sin dueño	9 %	15 %
suscripciones y licencias	6 %	10 %
	62 %	

Y cómo se atacó cada uno:

sin etiqueta
etiqueta obligatoria en la creación clase 253
y lo existente, por barrido con el inventario
→ 27 % → 2 % en 5 meses

- compartidos
 - red y registros repartidos por uso medido
 - plataforma, por número de servicios
 - regla ACORDADA en una reunión de 40 minutos
 - imperfecta y suficiente

cuentas antiguas
9 cuentas sin dueño; 4 se apagaron enteras
→ y no pasó nada, que era la sospecha

atribución total 38 % → 96 % en 7 meses

La unidad económica y la conversación que cambió.

se eligió con negocio: COSTE POR PEDIDO COMPLETADO
y una segunda: coste por cliente activo mensual

el histórico, reconstruido con 24 meses de datos

	factura/mes	pedidos/mes	coste/pedido
mes -24	198.000	412.000	0,48
mes -18	241.000	506.000	0,48
mes -12	287.000	648.000	0,44
mes -6	319.000	741.000	0,43
mes 0	341.000	832.000	0,41

Y lo que esa tabla hizo con la discusión:

la factura había subido un 72 % en dos años
y el coste por pedido había bajado un 15 %

- dirección dejó de preguntar por la factura
- y empezó a preguntar por la tendencia de la unidad
- y la pregunta pasó a ser «¿cuánto más se puede bajar y qué cuesta bajarlo?»

Las palancas, por orden de rendimiento real.

1 TRABAJO QUE NO HACÍA FALTA

sondeo del estado del pedido desde el navegador	
→ eventos	clase 210
-22 % de peticiones	-31.000 USD/mes
registros de acceso conservados 400 días	
→ 30 días en caliente, 13 meses en frío	
	-19.400 USD/mes
entornos de ensayo encendidos 24 horas	
→ apagado nocturno automático	
	-11.200 USD/mes
tres copias del mismo conjunto de datos en tres equipos	
→ un producto de datos con contrato	clase 241
	-8.900 USD/mes
subtotal	
	-70.500 USD/mes

2 FORMA DE LOS DATOS

informes sobre ficheros de texto → columnar y
particionado clase 243
coste del informe diario 340 → 12 USD
y el conjunto de 41 informes
-37.800 USD/mes

3 MOVER TRABAJO

```
reprocesos y entrenamientos a capacidad interrumpible
-14.100 USD/mes
```

4 AJUSTE DE TAMAÑOS

-9.600 USD/mes

5 COMPROMISOS, aplicados al final
sobre la base ya reducida y estable
-22.400 USD/mes

Y la observación que el equipo destacó:

el ajuste de tamaños –lo que casi todas las herramientas recomiendan primero– fue la CUARTA palanca y aportó el 6 % del total

y el trabajo innecesario aportó el 45 %
→ y no lo detecta ninguna herramienta de coste, porque desde la factura ese gasto parece legítimo
→ hace falta mirar el sistema, no la factura

Y el error que estuvo a punto de cometerse:

en el mes 2, finanzas propuso firmar compromisos a 3 años sobre el consumo de ese momento
descuento ofrecido -31 %

se retrasó a después de las optimizaciones
consumo tras optimizar -46 %

→ el compromiso sobre el consumo original habría congelado una arquitectura que se iba a reducir a la mitad
→ y habría obligado a pagar capacidad no usada durante 3 años
→ el descuento del 31 % sobre el doble de consumo es peor que el 22 % sobre la mitad

Año 2 · Cuando el ahorro obvio se agotó.

situación	
factura	341.000 → 227.000 USD/mes
coste por pedido	0,41 → 0,17
pedidos	832.000 → 1.114.000

y la pregunta de dirección
«¿qué más vais a ahorrar este año?»
→ y la respuesta honesta era «poco»

→ y aquí es donde la función suele desaparecer

Y lo que la salvó: el registro de coste evitado.

durante 14 meses, cada decisión de arquitectura con implicación de coste se había registrado con
la alternativa considerada
el coste estimado de cada una
y quién decidió

lo acumulado en el año 2

decisión	alternativa	evitado/año
objetivo de recuperación de 4 h en vez de activo-activo	2ª región	1.344.000
partición por fecha desde el diseño del almacén	exploración completa	394.000
eventos en vez de sondeo en el servicio nuevo de envíos	sondeo	78.000
caché en el borde para el catálogo	más réplicas	61.000
retención de 30 días desde el principio en el servicio de recomendaciones	400 días	44.000
... (11 decisiones más)		218.000
		2.139.000

Y cómo se presentó:

no como «hemos ahorrado 2,1 millones»
→ eso sería falso: ese gasto nunca existió

sino así

«en 14 meses, 16 decisiones de arquitectura se tomaron con la cifra delante. Las alternativas descartadas habrían costado 2,1 M USD al año. Cada decisión está registrada con quién la tomó.»

→ y la función pasó de «los que recortan» a «los que ponen la cifra antes de decidir»

→ que es exactamente el cambio que define el nivel 4

Las cifras a los dos años.

coste por pedido	0,41	0,17
coste por cliente activo	2,14	1,03
factura mensual	341.000	227.000
pedidos mensuales	832.000	1.114.000
coste atribuido	38 %	96 %
recursos sin dueño	1.107	12
decisiones de diseño con coste estimado antes	0 %	84 %
coste evitado registrado (año)	-	2.139.000
error de la previsión mensual	21 %	3,4 %
reuniones trimestrales sobre «por qué sube la nube»	4	0

La lección que esta clase deja: el ajuste de tamaños —lo primero que recomienda cualquier herramienta— aportó el 6 % del resultado, y eliminar trabajo que no hacía falta aportó el 45 %, porque desde la factura ese gasto parece legítimo y solo se ve mirando el sistema. Y cuando el ahorro obvio se agotó, lo que sostuvo la función no fue lo recortado sino 2,1 millones al año de alternativas descartadas, registradas en el momento de decidir y con nombre de quien decidió.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-22-specializations-certifications-career/270-ruta-finops-practitioner/lab.py
```

El laboratorio selecciona el motor de práctica decision y produce labresult.json. El escenario, sus comprobaciones y el artefacto esperado corresponden a esta clase; no requiere credenciales y deja explícito qué debe revalidarse en un sandbox real.

1. Lee exercise.steps y formula una predicción antes de ejecutar.
2. Ejecuta la práctica y verifica todos los elementos de checks.
3. Provoca el caso de negativetest y explica la señal observada.
4. Materializa finops-plan en evidence/ usando la plantilla indicada.
5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

El artefacto principal es una matriz de decisión ponderada y un ADR. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye finops-plan para el caso CloudShop. Incluye una alternativa descartada, un supuesto que pueda falsarse, una prueba de fallo y una decisión de rollback.

■ Criterio de aceptación

- [] lab.py termina con código 0 y genera JSON válido.
- [] La entrega conecta al menos tres requisitos con mecanismos verificables.
- [] Existe una prueba positiva y una prueba negativa con evidencia.
- [] Seguridad, costo y operación aparecen como decisiones, no como anexos.

- [] Se declara una limitación y una condición que obligaría a revisar el diseño.
- [] Otra persona puede repetir el recorrido sin conocimiento tácito.

■ Errores frecuentes

Síntoma	Causa probable	Corrección
El ahorro obvio se agota y la función parece innecesaria	Se midió por ahorro conseguido, que no ve el coste evitado	Registra cada decisión de diseño con su alternativa y coste estimado en el momento de decidir; ahí está la mayor parte del valor.
Se atribuye el coste a los equipos y nadie lo reduce	Se atribuyó sin dar palanca: el equipo no puede cambiar lo que le cuesta	Atribuye a quien puede decidir y acompaña la atribución con la capacidad de cambiarlo.
Se firman compromisos y luego sobra capacidad pagada	Se comprometió antes de optimizar, congelando la arquitectura que iba a cambiar	Aplica compromisos al final, sobre una base ya reducida y estable; el descuento sobre el doble de consumo es peor negocio.
Hay paneles de coste completos y el gasto sigue creciendo	El coste se ve en el informe mensual, no donde se toma la decisión	Pon el coste estimado en la propuesta de cambio antes de aprobarla y añade alertas de anomalía por servicio con dueño.
Las herramientas recomiendan ajustar tamaños y el efecto es pequeño	El desperdicio real es trabajo innecesario, que desde la factura parece legítimo	Empieza por sondeos, reintentos, copias duplicadas, retención excesiva y entornos encendidos; mira el sistema, no la factura.
Cada trimestre se discute por qué sube la nube	Se habla de factura absoluta en vez de coste por unidad de negocio	Define la unidad económica con negocio y presenta su tendencia; una factura creciente con unidad decreciente es una señal sana.

■ Seguridad, ética y costo

Trabaja con cuentas propias o sandboxes autorizados. No publiques secretos, identificadores reales ni datos personales. Antes de crear recursos pagos define presupuesto, etiquetas y comando de destrucción; después verifica que no queden recursos huérfanos. Los ejemplos locales enseñan contratos, pero no certifican cumplimiento ni disponibilidad de producción.

■ Preguntas de comprobación

1. ¿Qué diferencia recortar factura de gestionar el coste del sistema?
2. ¿Por qué la factura es un mal punto de intervención?
3. ¿Qué revela la unidad económica que el coste absoluto oculta?
4. ¿Qué es el coste evitado y cómo se registra para que sea creíble?
5. ¿Por qué medirse por el ahorro conseguido es un modo de fracaso?

Referencias

- Storment, J. R. y Fuller, M. (2023). Cloud FinOps, 2.^a ed.
<<https://www.oreilly.com/library/view/cloud-finops-2nd/9781492098348/>>
- FinOps Foundation (2024). FinOps Framework: capabilities and maturity. <<https://www.finops.org/framework/>>
- AWS (2024). Cost Optimization Pillar, Well-Architected Framework.
<<https://docs.aws.amazon.com/wellarchitected/latest/cost-optimization-pillar/welcome.html>>
- Microsoft (2024). Azure Well-Architected Framework: cost optimization.
<<https://learn.microsoft.com/azure/well-architected/cost-optimization/>>
- Google Cloud (2024). Cost management and FinOps best practices.
<<https://cloud.google.com/architecture/framework/cost-optimization>>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Anterior	Índice	Siguiente
← 269 · Ruta Cloud Security Engineer	Parte 22 · Programa	271 · Ruta Cloud Data y AI Engineer →

Evaluación — 270 Ruta FinOps Practitioner

Preguntas

1. Explica el problema que resuelve esta capacidad sin mencionar un proveedor.
2. Identifica una frontera de responsabilidad y su propietario.
3. Compara dos alternativas mediante confiabilidad, seguridad, costo y operación.
4. ¿Qué demuestra el laboratorio y qué no puede demostrar?
5. Propón un caso límite o una prueba de fallo.

Reto verificable

Entrega finops-plan con diagrama o configuración, evidencia de ejecución, alternativa descartada, riesgo residual y criterio de rollback.

Criterio de aceptación

- ☐ Resultado reproducible por una segunda persona.
- ☐ Evidencia vinculada a cada afirmación técnica.
- ☐ Prueba negativa o de fallo incluida.
- ☐ Costos y permisos expresados con unidades y alcance.
- ☐ Limitaciones declaradas sin presentar la simulación como producción.

Escala

Nivel	Evidencia
A — competente	Cumple todo y defiende trade-offs con datos.
B — en desarrollo	Cumple lo esencial; falta profundidad en una dimensión.
C — inicial	Artefacto parcial o conclusión sin evidencia suficiente.
0	Sin entrega reproducible.

271 — Ruta Cloud Data y AI Engineer

Parte: 22 — Especializaciones, certificaciones y práctica profesional Nivel: intermedio-avanzado · Horas estimadas: 4
Laboratorio: decision · Estado: EXECUTABLECORE

Propósito

La ruta de datos e inteligencia artificial en la nube: conseguir que el dato sea confiable y que el modelo sirva en producción. La clase separa las tres funciones que suelen confundirse —ingeniería de datos, analítica y aprendizaje automático—, da las competencias que se miden en cada una, y marca el modo de fracaso que la parte 20 demostró con cifras: construir el modelo antes de que el dato sea de fiar.

Resultados de aprendizaje

Al finalizar podrás:

1. Separar ingeniería de datos, analítica y aprendizaje automático.
2. Reconocer qué competencias se miden en cada función y en qué orden.
3. Aplicar el orden por coste de cambio: contratos primero, modelo al final.
4. Detectar el modo de fracaso de modelar sobre datos no fiables.
5. Elegir entre las tres funciones según lo que se quiere resolver.

Conceptos centrales

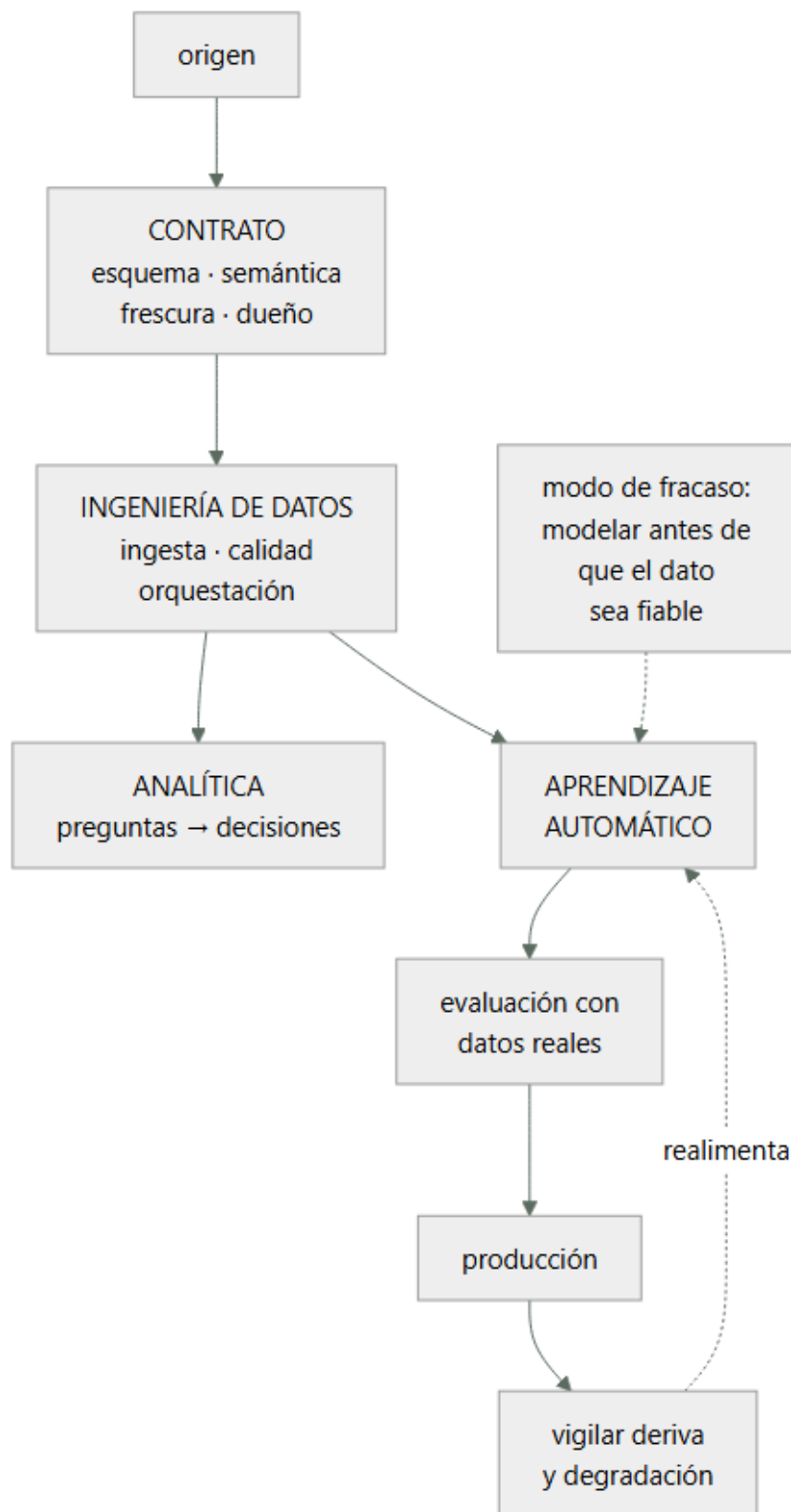
Concepto	Comprensión verificable
ingeniería de datos	Hacer que el dato llegue completo, a tiempo y con significado declarado. La base de las otras dos.
analítica	Convertir datos en respuestas que alguien usa para decidir. Su producto es la decisión, no el panel.
aprendizaje automático	Producir predicciones útiles en producción, con su evaluación y su degradación vigiladas.
contrato de datos	Acuerdo escrito sobre esquema, semántica, frescura y dueño. Lo más caro de cambiar.
sesgo entrenar-servir	Calcular un atributo distinto al entrenar y al servir. El fallo silencioso más común.
evaluación de producción	Medir la calidad con datos reales y en el sistema real, no con casos imaginados.

Modelo mental

Una especialización combina fundamentos, evidencia de proyectos y juicio bajo restricciones; una insignia sin práctica no sustituye esa combinación.

Aplicado a esta clase, separa siempre cuatro planos: intención (qué necesita el usuario), configuración (qué declaramos), estado observado (qué existe de verdad) y evidencia (cómo sabemos que cumple). Confundirlos produce diseños que se ven correctos en un diagrama pero fallan al operar.

■ Flujo de razonamiento



Desarrollo

1. Tres funciones distintas

Se anuncian juntas y exigen cosas diferentes. Confundirlas produce contrataciones frustradas en las dos direcciones.

INGENIERÍA DE DATOS
que el dato llegue completo, a tiempo y con significado
declarado

→ ingesta, orquestación, calidad, formatos, particiones,
reproceso clases 241-244
su producto conjuntos de datos en los que otros confían
su fallo nadie se entera hasta semanas después
ley 29

ANALÍTICA

convertir datos en respuestas que alguien usa
→ modelado dimensional, métricas acordadas, paneles
su producto DECISIONES, no paneles
su fallo paneles preciosos que nadie mira, o dos
cifras distintas para lo mismo

APRENDIZAJE AUTOMÁTICO

producir predicciones útiles en producción
→ atributos, entrenamiento, servicio, evaluación,
deriva clases 244-250
su producto un sistema que decide mejor que la regla
que sustituye
su fallo funciona en el laboratorio y no en
producción

→ y las tres dependen de la primera
→ que es lo que la parte 20 demostró midiendo

Y la jerarquía que no se puede saltar:

SIN CONTRATO NO HAY DATO FIABLE
SIN DATO FIABLE NO HAY MÉTRICA COMPARABLE
SIN MÉTRICA COMPARABLE NO HAY MODELO EVALUABLE

→ y el error de la parte 20, medido: un campo llamado
«importe» que significaba tres cosas distintas y cuyo
esquema era idéntico en las tres tablas
→ el error más caro no lo cometió ningún modelo

Y el orden por coste de cambio, que ordena cualquier proyecto:

- 1 contratos y semántica lo más caro de cambiar
- 2 ingesta y reproceso
- 3 calidad que DETIENE clase 243
- 4 atributos, con el mismo cálculo al entrenar y al servir
- 5 servicio y evaluación
- 6 y el modelo lo más barato de cambiar

→ y casi todos los proyectos empiezan por el 6

2. Las competencias que se miden

Por función y por nivel, lo que separa a alguien que resuelve de alguien que ha leído.

INGENIERÍA DE DATOS

nivel 2
monta ingesta incremental con captura de cambios y
sabe qué pasa con los borrados clase 242
define marca de agua y trata lo tardío
elige formato y partición con criterio de coste
clase 243
y hace reproceso reproducible
nivel 3
diseña contratos con semántica escrita clase 241
decide la arquitectura de datos y sus fronteras
monta calidad que detiene antes de publicar
y baja el plazo de publicar un dato nuevo
nivel 4
el consumo pasa por la plataforma porque conviene, no
porque se obligue ley 16

ANALÍTICA

nivel 2
modela para consultar, no para almacenar
define métricas con su definición escrita
y sabe por qué dos paneles no coinciden
nivel 3
acuerda las métricas con negocio y las hace únicas

diseña para que la respuesta llegue cuando se decide
y retira paneles
nivel 4
las decisiones de la empresa usan las mismas cifras

APRENDIZAJE AUTOMÁTICO

nivel 2
entrena, evalúa y despliega un modelo
calcula atributos igual al entrenar y al servir
clase 244
y mide en producción, no solo en el laboratorio
nivel 3
diseña la evaluación con casos REALES clase 250
decide entre modelo y regla, con coste y riesgo
vigila deriva y degradación clase 246
y sabe cuándo NO usar un modelo
nivel 4
el sistema se evalúa continuamente y la organización
acepta retirar modelos que no aportan

Y las preguntas que más discriminan en una entrevista:

«¿cómo detectáis que un dato está mal si no da error?»
→ discrimina brutalmente ley 29

«¿qué pasa con los borrados en vuestra ingesta
incremental?»
→ la mayoría no lo ha pensado clase 242

«¿el atributo se calcula igual al entrenar y al servir?
¿cómo lo comprobáis?»

«¿de dónde salieron los casos de vuestra evaluación?»
→ si los imaginó el equipo, el número no significa nada
clase 250

y «¿qué modelo habéis retirado?»
→ nivel 3 ha retirado alguno

3. El modo de fracaso y sus señales

El modo de fracaso de esta ruta es empezar por el final, y tiene señales tempranas.

MODELAR ANTES DE QUE EL DATO SEA FIABLE
se construye el modelo, funciona en el laboratorio, y en
producción da resultados peores sin que nadie sepa por
qué

las causas, por frecuencia medida en la parte 20
el atributo se calcula distinto al entrenar y al servir
clase 244
el conjunto de entrenamiento contiene información que
en producción no existe todavía
la evaluación se hizo con casos imaginados
clase 250
el dato de entrada cambió de significado ley 29
y la distribución se movió sin que nadie mirara
clase 246

→ las cinco son de DATOS, no de modelo
→ y ninguna da error

Y las señales tempranas, que se pueden mirar hoy:

- ¿existe una definición escrita de cada campo que usamos?
- ¿sabemos cuándo se actualizó por última vez cada conjunto?
- ¿hay comprobaciones que DETIENEN la publicación?
- ¿los atributos se calculan en un solo sitio?
- ¿la evaluación usa casos que ocurrieron de verdad?
- ¿alguien mira la distribución además del volumen?

→ si más de dos son «no», el modelo va a fallar y la causa no será el modelo

Y los otros dos modos de fracaso de la ruta:

EL ALMACÉN QUE NADIE USA

se construye la plataforma de datos y el consumo la rodea con exportaciones y copias
→ y en la parte 20 eso era el 71 %
la causa medida: publicar por el camino oficial tardaba once semanas
→ lo que lo revirtió fue bajarlo a dos días, no una norma ley 16

EL MODELO QUE NADIE PUEDE RETIRAR

entra en producción, nadie mide si aporta, y quitarlo da miedo
→ la defensa es medir el efecto contra la alternativa simple DESDE EL PRINCIPIO
→ y muchas veces la regla de tres líneas gana

Y la pregunta que ahorra proyectos enteros:

«¿QUÉ DECISIÓN VA A CAMBIAR CON ESTO, Y QUIÉN LA TOMA?»

→ si no hay respuesta, no hay proyecto
→ y esto vale igual para un panel que para un modelo

y la segunda

«¿CUÁL ES LA ALTERNATIVA SIMPLE Y CUÁNTO DE LO PROMETIDO CONSIGUE?»

→ una media móvil, una regla, un umbral
→ y con frecuencia consigue el 80 % con el 2 % del coste

4. Elegir, demostrar y el techo

Cómo se decide entre las tres funciones y cómo se demuestra la ruta.

ELIGE INGENIERÍA DE DATOS SI

te interesan sistemas, fiabilidad y volumen
toleras que tu trabajo sea invisible cuando funciona
y te motiva que otros construyan encima
→ es la que más demanda tiene y la que menos se anuncia

ELIGE ANALÍTICA SI

te interesa el negocio tanto como la técnica
te motiva discutir definiciones hasta que sean únicas
y aceptas que tu producto es la decisión ajena

ELIGE APRENDIZAJE AUTOMÁTICO SI

toleras la incertidumbre y la evaluación constante
y aceptas que la mayor parte del trabajo son datos y producción, no modelado
→ y quien entra esperando lo contrario se va

Y la evidencia que vale:

LO QUE NO VALE

«monté un canal de datos con estas herramientas»
«entrené un modelo con un 94 % de acierto»
→ el segundo, además, es sospechoso: ¿medido con qué?

LO QUE VALE

«el plazo de publicar un dato nuevo pasó de once semanas a dos días, y el consumo que pasa por la plataforma subió del 29 % al 96 %»
«detectamos un cambio de céntimos a euros que había pasado todas las comprobaciones de validez, y añadimos la comparación de distribución que lo habría cogido»
«el conjunto de evaluación con preguntas reales dio 67 % frente al 94 % del imaginado; con eso rediseñamos la recuperación»
«retiramos el modelo de recomendación porque una regla conseguía el 91 % del efecto con el 3 % del coste»

→ efecto, mecanismo y cifra clase 275

Y el techo:

EL TECHO

los datos son fiables, las métricas únicas y los modelos se evalúan y se retiran
→ y lo que limita entonces es qué decide la organización con ellos

continuaciones

- a ARQUITECTURA clase 272
si el límite es cómo están organizados los sistemas
- b producto o dirección de datos
si el límite es qué preguntas se hacen
- c o plataforma clase 267
si lo que falta es que publicar un dato sea fácil

Y la lista de comprobación de la clase:

- sé cuál de las tres funciones estoy haciendo
- no empiezo por el modelo
- cada campo que uso tiene definición escrita y dueño
- hay comprobaciones que detienen la publicación
- los atributos se calculan en un solo sitio
- la evaluación usa casos que ocurrieron de verdad
- alguien mira la distribución, no solo el volumen
- mido el efecto contra la alternativa simple
- sé qué decisión cambia con lo que construyo y quién la toma
- publicar un dato nuevo se mide en días, no en semanas
- he retirado algún panel o algún modelo

Y el cierre que enlaza con la clase siguiente: queda la ruta que no construye ninguna pieza y responde de que las decisiones sean defendibles y sobrevivan a quien las tomó. Arquitectura de soluciones es la materia de la clase 272.

Ejemplo trabajado

Tres trayectorias en la ruta de datos, con lo que cada una midió. Lo que sigue es el proyecto que empezó por el modelo y tuvo que rehacerse, la analista que retiró 61 paneles, y el ingeniero de datos que bajó un plazo de once semanas a dos días.

Caso 1 · El proyecto que empezó por el modelo.

encargo predecir la fecha de entrega de un pedido
equipo 2 personas, 4 meses

lo que se hizo, en este orden

- 1 se recogieron datos históricos con una consulta
- 2 se entrenaron seis modelos
- 3 se eligió el mejor
- 4 se desplegó

resultado en el laboratorio error medio 1,4 días
resultado en producción error medio 3,9 días

Y el diagnóstico, que tardó cinco semanas:

- causa 1 17 atributos se calculaban con una consulta al entrenar y con código distinto al servir
→ y 6 de ellos daban valores diferentes
clase 244
- causa 2 el conjunto de entrenamiento incluía el estado final del pedido, que al predecir no existe
→ información del futuro
- causa 3 el campo «fecha de envío» significaba «fecha planificada» en una tabla y «fecha real» en otra
→ y el esquema era idéntico

→ las tres son de datos
→ y el modelo era correcto

Y lo que costó rehacerlo en el orden correcto:

contratos y semántica de 9 campos 3 semanas
→ la discusión sobre «fecha de envío» duró 4 días
una sola definición de cada atributo,
usada al entrenar y al servir 2 semanas

comprobaciones que detienen	1 semana
evaluación con pedidos reales del último trimestre	1 semana
el modelo, reentrenado	2 días

resultado en producción error medio 1,6 días

→ el modelo fue lo último y lo más rápido
→ y las 7 semanas de datos eran inevitables: solo se pagaron dos veces

Caso 2 · La analista que retiró 61 paneles.

situación	
paneles existentes	94
con más de una visita al mes	33
con más de una visita a la semana	19

y el problema real
«pedidos completados» tenía cuatro definiciones distintas en cuatro paneles
incluye o no cancelados posteriormente
cuenta por fecha de pedido o de pago
incluye o no pedidos de prueba
y una usaba zona horaria local y otra universal

→ y en una reunión de dirección se presentaron dos cifras del mismo mes con un 11 % de diferencia

Y el trabajo, que fue de acuerdo más que de técnica:

- 1 se listaron las métricas usadas en decisiones reales
→ 14, de las 240 definidas
- 2 cada una se acordó con su dueño de negocio
→ definición escrita, con los casos límite resueltos
→ «pedido completado: pagado y no cancelado en 30 días, por fecha de pago, en zona horaria universal, excluyendo cuentas de prueba»
- 3 se implementaron una sola vez, en una capa común
- 4 y los paneles se reconstruyeron sobre ellas
- 5 se retiraron 61 paneles
→ y se avisó con 4 semanas; 3 personas reclamaron; se recuperaron 2

resultado	
métricas con definición única y escrita	14
paneles	94 → 33
discrepancias en reuniones de dirección	11 → 0
tiempo de la analista en «¿por qué no cuadra?»	9 h/semana → 1

Y lo que la analista escribió en su revisión anual:

«mi trabajo del año no fue construir nada: fue conseguir que catorce cifras significaran una sola cosa, y retirar sesenta y un paneles»

→ y esa frase describe el nivel 3 de analítica mejor que cualquier lista de herramientas

Caso 3 · De once semanas a dos días.

situación	
publicar un conjunto de datos nuevo en la plataforma	
tardaba 11 semanas	

el desglose, medido con 6 casos	
solicitud y aprobación	2,1 semanas
revisión de esquema por el equipo central	3,4
implementación del canal	2,8
revisión de seguridad y permisos	1,9
documentación y publicación	0,8

y la consecuencia
el 71 % del consumo de datos rodeaba la plataforma

- exportaciones, accesos directos y copias
- cada una sin contrato, sin calidad y sin dueño

Y lo que se cambió:

la aprobación se sustituyó por criterios automáticos
 → 2,1 semanas → 0
 la revisión de esquema pasó a una comprobación automatizada del contrato, con revisión humana solo si falla
 → 3,4 semanas → 20 minutos en el 84 % de los casos
 la implementación del canal pasó a plantilla
 → 2,8 semanas → 4 horas
 los permisos se derivaron del contrato
 → 1,9 semanas → automático
 y la documentación se generó del contrato
 → 0,8 semanas → automático

plazo total 11 semanas → 2 días

Y el efecto, doce meses después:

plazo de publicación	11 semanas	2 días
conjuntos publicados por trimestre	3	41
consumo que pasa por la plataforma	29 %	96 %
copias no gobernadas detectadas	34	3
incidentes por dato incorrecto	19/año	4/año

y el coste del trabajo 340 horas

Y la observación que se llevó a la dirección:

no se prohibió nada
 → el consumo volvió a la plataforma porque el camino oficial pasó a ser el más rápido
 → ley 16, aplicada en la dirección correcta

 y la comparación con el intento anterior
 18 meses antes se había emitido una NORMA que obligaba a publicar por la plataforma
 → consumo por la plataforma tras la norma: 29 % → 31 %
 → tras bajar el plazo: 96 %

La lección que esta clase deja: el proyecto que empezó por el modelo tuvo un error de 3,9 días en producción frente a 1,4 en el laboratorio, y las tres causas eran de datos —atributos calculados distinto, información del futuro y un campo que significaba dos cosas—, ninguna del modelo. Y una norma que obligaba a usar la plataforma movió el consumo del 29 % al 31 %; bajar el plazo de once semanas a dos días lo movió al 96 %.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-22-specializations-certifications-career/271-ruta-cloud-data-y-ai-engineer/lab.py
```

El laboratorio selecciona el motor de práctica decision y produce labresult.json. El escenario, sus comprobaciones y el artefacto esperado corresponden a esta clase; no requiere credenciales y deja explícito qué debe revalidarse en un sandbox real.

1. Lee exercise.steps y formula una predicción antes de ejecutar.
2. Ejecuta la práctica y verifica todos los elementos de checks.
3. Provoca el caso de negativetest y explica la señal observada.
4. Materializa data-ai-plan en evidence/ usando la plantilla indicada.
5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

El artefacto principal es una matriz de decisión ponderada y un ADR. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye data-ai-plan para el caso CloudShop. Incluye una alternativa descartada, un supuesto que pueda falsarse, una prueba de fallo y una decisión de rollback.

■ Criterio de aceptación

- [] lab.py termina con código 0 y genera JSON válido.
- [] La entrega conecta al menos tres requisitos con mecanismos verificables.
- [] Existe una prueba positiva y una prueba negativa con evidencia.
- [] Seguridad, costo y operación aparecen como decisiones, no como anexos.
- [] Se declara una limitación y una condición que obligaría a revisar el diseño.
- [] Otra persona puede repetir el recorrido sin conocimiento tácito.

■ Errores frecuentes

Síntoma	Causa probable	Corrección
El modelo funciona en el laboratorio y falla en producción	Los atributos se calculan distinto al entrenar y al servir, o el entrenamiento vio información que al predecir no existe	Calcula cada atributo en un solo sitio, comprueba la equivalencia y revisa que ningún campo del conjunto contenga estado posterior a la predicción.
Dos paneles dan cifras distintas para lo mismo	La métrica tiene varias definiciones y ninguna está escrita	Acuerda con negocio una definición única con los casos límite resueltos, impleméntala una vez en capa común y reconstruye encima.
La plataforma de datos existe y el consumo la rodea	Publicar por el camino oficial tarda semanas	Mide el plazo por tramos y automatiza aprobación, revisión de esquema, canal y permisos; el camino oficial tiene que ser el más rápido.
La evaluación da un número excelente y los usuarios se quejan	Los casos de evaluación los imaginó el equipo	Construye el conjunto con casos que ocurrieron de verdad; si el número cae mucho, ese es el número real.
Un dato incorrecto se descubre semanas después	Los fallos de datos no dan error: dan otro número	Añade comprobaciones que detengan la publicación y compara distribución, volumen, frescura y completitud contra el histórico.
Hay un modelo en producción que nadie sabe si aporta	No se midió contra la alternativa simple desde el principio	Compara siempre con una regla o un umbral; muchas veces consigue la mayor parte del efecto por una fracción del coste, y entonces se retira el modelo.

■ Seguridad, ética y costo

Trabaja con cuentas propias o sandboxes autorizados. No publiques secretos, identificadores reales ni datos personales. Antes de crear recursos pagos define presupuesto, etiquetas y comando de destrucción; después verifica que no queden recursos huérfanos. Los ejemplos locales enseñan contratos, pero no certifican cumplimiento ni disponibilidad de producción.

■ Preguntas de comprobación

1. ¿Qué distingue ingeniería de datos, analítica y aprendizaje automático?
2. ¿Por qué el modelo debe ser lo último y no lo primero?
3. ¿Qué señales tempranas indican que un modelo va a fallar por datos?
4. ¿Qué pregunta ahorra proyectos enteros antes de empezarlos?
5. ¿Qué evidencia demuestra nivel 3 en cada una de las tres funciones?

Referencias

- Kleppmann, M. (2017). Designing Data-Intensive Applications. <<https://dataintensive.net/>>
- Sculley, D. y otros (2015). Hidden technical debt in machine learning systems. <<https://papers.nips.cc/paper/2015/hash/86df7dcfd896fcac2674f757a2463eba-Abstract.html>>
- Google Cloud (2024). MLOps: continuous delivery and automation pipelines. <<https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning>>
- AWS (2024). Machine Learning Lens, Well-Architected Framework. <<https://docs.aws.amazon.com/wellarchitected/latest/machine-learning-lens/machine-learning-lens.html>>
- Microsoft (2024). Azure MLOps and data governance guidance. <<https://learn.microsoft.com/azure/architecture/ai-ml/guide/machine-learning-operations-v2>>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 22 en PDF · Manual integral

Anterior	Índice	Siguiente
← 270 · Ruta FinOps Practitioner	Parte 22 · Programa	272 · Ruta Cloud Solutions Architect →

Evaluación — 271 Ruta Cloud Data y AI Engineer

Preguntas

1. Explica el problema que resuelve esta capacidad sin mencionar un proveedor.
2. Identifica una frontera de responsabilidad y su propietario.
3. Compara dos alternativas mediante confiabilidad, seguridad, costo y operación.
4. ¿Qué demuestra el laboratorio y qué no puede demostrar?
5. Propón un caso límite o una prueba de fallo.

Reto verificable

Entrega data-ai-plan con diagrama o configuración, evidencia de ejecución, alternativa descartada, riesgo residual y criterio de rollback.

Criterio de aceptación

- [] Resultado reproducible por una segunda persona.
- [] Evidencia vinculada a cada afirmación técnica.
- [] Prueba negativa o de fallo incluida.
- [] Costos y permisos expresados con unidades y alcance.
- [] Limitaciones declaradas sin presentar la simulación como producción.

Escala

Nivel	Evidencia
A — competente	Cumple todo y defiende trade-offs con datos.
B — en desarrollo	Cumple lo esencial; falta profundidad en una dimensión.
C — inicial	Artefacto parcial o conclusión sin evidencia suficiente.
0	Sin entrega reproducible.

272 — Ruta Cloud Solutions Architect

Parte: 22 — Especializaciones, certificaciones y práctica profesional Nivel: intermedio-avanzado · Horas estimadas: 4
Laboratorio: decision · Estado: EXECUTABLECORE

Propósito

La ruta de arquitectura: responder de que las decisiones sean defendibles y sobrevivan a quien las tomó. La clase da lo que esta especialidad produce de verdad —decisiones escritas con sus alternativas y su reversibilidad—, el método para tomarlas, y sus dos modos de fracaso: la arquitectura de diapositivas, que no toca el sistema, y la arquitectura de control, que aprueba en vez de decidir.

Resultados de aprendizaje

Al finalizar podrás:

1. Producir decisiones escritas con alternativas, criterio y consecuencias.
2. Clasificar decisiones por reversibilidad para ajustar el rigor.
3. Defender una decisión ante quien no es técnico, con cifras.
4. Evitar la arquitectura que no toca el sistema y la que solo aprueba.
5. Reconocer qué hace que una decisión sobreviva a su autor.

Conceptos centrales

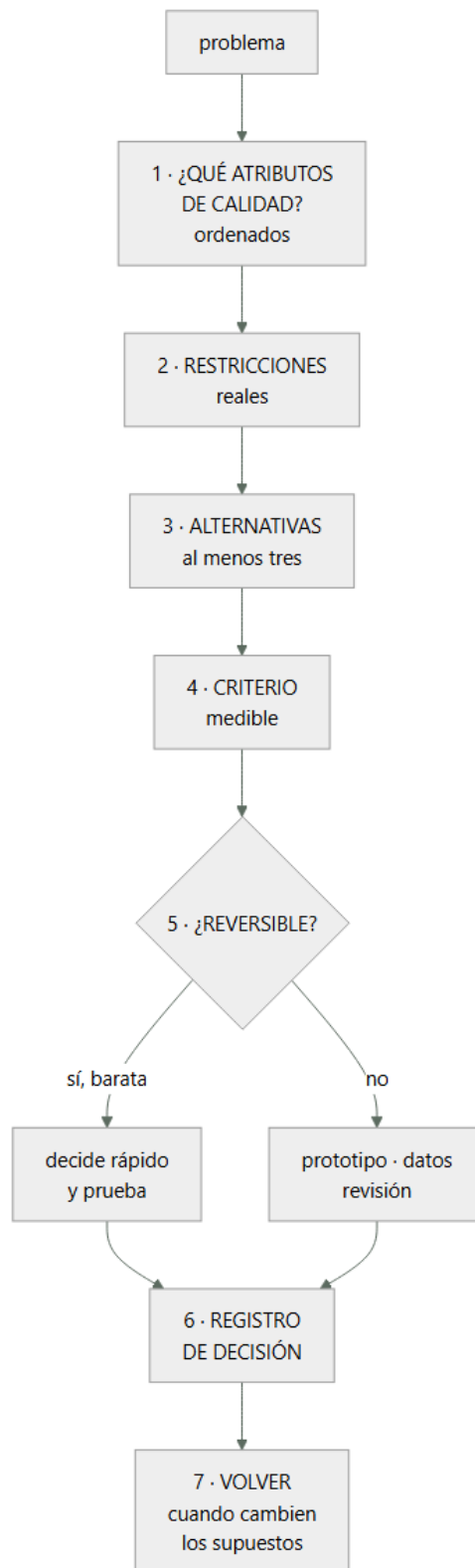
Concepto	Comprensión verificable
registro de decisión	Documento breve con contexto, alternativas, decisión, criterio y consecuencias. El producto de esta ruta.
reversibilidad	Cuánto cuesta deshacer una decisión. Determina cuánto rigor merece.
atributo de calidad	Propiedad del sistema que se prioriza: latencia, disponibilidad, coste, evolución, seguridad.
compromiso explícito	Decir qué se empeora al mejorar otra cosa. Sin él no hay decisión, hay preferencia.
arquitectura de diapositivas	Modo de fracaso en que se diseña sin construir y el sistema real diverge del diagrama.
restricción real	Lo que el sistema no puede saltarse: plazos, normativa, personas disponibles, dinero.

Modelo mental

Una especialización combina fundamentos, evidencia de proyectos y juicio bajo restricciones; una insignia sin práctica no sustituye esa combinación.

Aplicado a esta clase, separa siempre cuatro planos: intención (qué necesita el usuario), configuración (qué declaramos), estado observado (qué existe de verdad) y evidencia (cómo sabemos que cumple). Confundirlos produce diseños que se ven correctos en un diagrama pero fallan al operar.

■ Flujo de razonamiento



Desarrollo

1. Qué produce esta ruta

No produce diagramas ni sistemas: produce decisiones que otros pueden entender, cuestionar y revisar cuando cambien los supuestos.

EL REGISTRO DE DECISIÓN, que es el entregable
CONTEXTO qué problema y con qué restricciones

ATRIBUTOS	qué priorizamos y en qué orden
ALTERNATIVAS	al menos tres, con lo bueno y lo malo de cada una
DECISIÓN	cuál y por qué
CRITERIO	qué medida la justifica
CONSECUENCIAS	qué empeora y qué nos obliga a hacer
REVISIÓN	qué supuesto, si cambia, invalida esto

- una o dos páginas, no veinte
- y lo que lo hace útil es la última línea: la revisión

Y por qué el registro importa más que el diagrama:

UN DIAGRAMA DICE QUÉ HAY
UN REGISTRO DICE POR QUÉ, Y CUÁNDO DEJARÁ DE VALER

- y a los dos años, quien mantiene el sistema no necesita saber qué hay: lo ve
- necesita saber por qué, para poder cambiarlo sin miedo
- y sin eso, cada decisión antigua se trata como sagrada o se rompe sin entenderla

Y los atributos de calidad, que hay que ordenar:

latencia · disponibilidad · coste · evolución ·
seguridad · operabilidad · coherencia

- y no se pueden maximizar todos
- decir «todos son importantes» es no decidir
- el trabajo es ORDENARLOS con quien tiene el problema

y la pregunta que fuerza el orden
«si tuviéramos que empeorar uno para mejorar otro,
¿cuál empeoramos?»
→ y esa conversación es el 70 % del trabajo

Y el compromiso explícito, sin el cual no hay decisión:

«elegimos una base relacional única»
→ mejora coherencia y simplicidad operativa
→ empeora escalado de escritura y aislamiento por servicio
→ y nos obliga a vigilar el codo del recurso limitante
clase 262

- si no se puede nombrar lo que empeora, no se ha entendido la alternativa

2. El método, ajustado a la reversibilidad

No todas las decisiones merecen el mismo esfuerzo. La variable que lo determina es cuánto cuesta deshacerlas.

DECISIONES REVERSIBLES BARATAS
qué biblioteca, qué formato de registro, qué umbral
→ decidir rápido, probar y corregir
→ y gastar dos semanas en analizarlas es desperdicio

DECISIONES CARAS DE DESHACER

modelo de datos y particiones	clase 208
fronteras entre servicios	clase 106
proveedor de identidad	clase 209
estrategia entre regiones	clase 187
y el modelo de cuentas y aislamiento	clase 219

→ prototipo, datos y revisión por pares

- y el error más común es tratarlas al revés: analizar mucho lo barato y decidir por costumbre lo caro

Y las restricciones reales, que se ignoran demasiado:

no son solo técnicas

cuánta gente hay y qué sabe hacer	ley 23
qué se puede mantener de guardia	
qué exige la normativa	clase 251
qué presupuesto hay	
y cuánto tiempo hay	

- una arquitectura correcta que el equipo no puede operar es una arquitectura incorrecta
- y esta es la lección de la parte 21 aplicada al diseño

Y el método completo:

- 1 entender el problema y quién lo tiene
- 2 ordenar atributos de calidad con esa persona
- 3 listar restricciones reales
- 4 generar al menos TRES alternativas
 - dos alternativas suelen ser «la que quiero» y una de paja
- 5 definir el criterio medible ANTES de comparar
- 6 clasificar por reversibilidad y ajustar el rigor
- 7 decidir, escribir y comunicar
- 8 y volver cuando el supuesto de revisión cambie

Y el paso que casi nadie da y que separa niveles:

VOLVER

«esta decisión asume menos de 2.000 pedidos por minuto; por encima, hay que revisarla»
→ y una alerta cuando se acerque

- una decisión con condición de revisión es viva
- una sin ella se convierte en herencia

3. Los dos modos de fracaso

Esta ruta tiene dos formas características de dejar de ser útil, y son opuestas.

MODO 1 · LA ARQUITECTURA DE DIAPOSITIVAS

se diseña, se presenta y no se toca el sistema
→ y el sistema real diverge del diagrama en semanas

las señales

el diagrama no coincide con lo desplegado
las decisiones no consideran restricciones de operación
y quien diseña no recibe alertas de lo que diseñó

la corrección

quien decide construye una parte
quien decide entra en la rotación de guardia
y las decisiones se validan con un prototipo

- es el mismo mecanismo de la ley 30: quien no ejecuta pierde el juicio que solo da ejecutar

MODO 2 · LA ARQUITECTURA DE CONTROL

el arquitecto aprueba diseños ajenos
→ cola de espera, lotes grandes y responsabilidad desplazada
→ y es el comité de cambios con otro nombre
clases 260, 266

la corrección

dar principios y ejemplos, no aprobaciones
revisar por invitación y en las decisiones caras
y medir cuántas decisiones se toman bien SIN ti

Y la señal de que la ruta funciona:

EQUIPOS QUE TOMAN BUENAS DECISIONES SIN CONSULTARTE

→ y consultan cuando la decisión es cara de deshacer

- y eso se consigue con principios escritos, ejemplos y registros anteriores accesibles
- no con revisiones obligatorias

Y los niveles de la ruta:

NIVEL 2 · RESUELVO

diseña un servicio completo con sus compromisos claros
escribe registros de decisión que otro entiende
y sabe estimar coste y capacidad de lo que propone

NIVEL 3 · DISEÑO

decide fronteras entre servicios y equipos
defiende ante negocio con cifras y con alternativas
hace reversible lo que puede y avisa de lo que no
y anticipa el modo de fallo y la operación

NIVEL 4 · CAMBIO EL SISTEMA

los principios están escritos y se usan sin ti
las decisiones caras se detectan antes de tomarse
y las decisiones antiguas se revisan cuando cambian los
supuestos, sin drama

4. Defender una decisión

La parte que más gente subestima: una decisión correcta que no se sostiene ante quien paga no se ejecuta.

ANTE NEGOCIO

no se habla de tecnología
se habla de
qué se consigue y cuándo
cuánto cuesta, de operación y de construcción
qué riesgo se asume y cuál se evita
y qué se pierde con la alternativa

y la cifra que más funciona
el coste de la alternativa descartada clase 270
→ «activo-activo cuesta 112.000 al mes; el objetivo de
recuperación de 4 horas nos cuesta 9.000 y cumple
lo que el negocio pidió»

ANTE INGENIERÍA

las alternativas, con lo bueno de cada una dicho en
serio
→ si la alternativa descartada se presenta como tonta,
nadie se cree la comparación
el criterio medible
y lo que empeora, dicho antes de que lo digan ellos

ANTE UN PANEL DE REVISIÓN clase 276

la pregunta más frecuente es «¿y si...?»
→ y la buena respuesta no es tener respuesta a todo
→ es «no lo consideramos; ¿cambiaría el criterio?»

Y los errores de defensa más caros:

- 1 DEFENDER LA DECISIÓN EN VEZ DEL CRITERIO
→ si alguien aporta un dato nuevo, la decisión debe
poder cambiar
→ defender la decisión convierte una revisión en una
discusión personal
- 2 NO DECIR LO QUE EMPEORA
→ y cuando aparece, se pierde la credibilidad de todo
lo demás
- 3 PRESENTAR CERTEZA DONDE HAY ESTIMACIÓN
→ «costará 40.000» frente a «entre 30.000 y 55.000,
según el volumen; y por encima de X hay que
revisarlo»
- 4 Y NO TENER LA ALTERNATIVA BARATA EN LA MESA
→ siempre hay que llevar la opción simple, aunque no
se recomienda
→ porque si no la llevas tú, la lleva otro

Y la lista de comprobación de la clase:

- mis decisiones están escritas con alternativas y
criterio
- cada registro dice qué empeora
- cada registro dice qué supuesto, si cambia, lo invalida
- ajusto el rigor a la reversibilidad
- genero al menos tres alternativas de verdad
- incluyo restricciones de personas y de operación
- lo que diseño lo construyo en parte y lo opero

- no apruebo diseños ajenos por sistema
- mido cuántas decisiones buenas se toman sin mí
- llevo la alternativa barata a la mesa
- definiendo el criterio, no la decisión
- y presento rangos donde hay estimación

Y el cierre que enlaza con la clase siguiente: las ocho rutas comparten la base y se apoyan en poder traducir entre proveedores. El mapeo entre las tres nubes, los contenedores y la disciplina de coste es la materia de la clase 273.

Ejemplo trabajado

Tres decisiones de arquitectura de CloudShop con su registro real. Lo que sigue es la que se defendió con el coste de la alternativa, la que se revisó dos años después porque el supuesto cambió, y la que se tomó mal por analizar lo barato y decidir lo caro por costumbre.

Decisión 1 · Estrategia entre regiones.

CONTEXTO

el flujo de compra debe seguir disponible si cae una región. Negocio pide «que no se caiga nunca».

ATRIBUTOS, ordenados con negocio

- 1 disponibilidad del flujo de compra
- 2 coste
- 3 operabilidad (equipo de 9 personas)
- 4 latencia

RESTRICCIONES

9 personas de plataforma; guardia de 6
presupuesto anual de infraestructura: 2,8 M USD
y ningún requisito normativo de residencia

ALTERNATIVAS

- A una región, copias en otra, restauración manual
recuperación 8-14 h · +0 USD/mes
- B una región activa, otra en frío con infraestructura
desplegada y datos replicados
recuperación 3-4 h · +9.000 USD/mes
- C activo-activo entre dos regiones
recuperación < 5 min · +112.000 USD/mes
y +2 personas para operarlo

CRITERIO

coste por hora de caída evitada, contra el valor de una
hora de caída medido por negocio: 31.000 USD

DECISIÓN

B, con objetivo de recuperación de 4 horas

CONSECUENCIAS

empeora: una caída de región cuesta ~4 h de servicio
nos obliga a: ensayar la conmutación cada trimestre y
mantener las cuotas de la región secundaria
clase 262

REVISIÓN

si el valor de una hora de caída supera los 90.000 USD,
o si el equipo pasa de 15 personas, se revisa C

Y cómo se defendió:

la presentación a dirección duró 11 minutos

lo que se dijo

«C cumple lo que pedís y cuesta 1,34 M al año más dos personas. Evita unas 7,7 horas de caída al año, que valen 239.000. B cuesta 108.000 al año y deja 4 horas de exposición. Recomendamos B y revisamos si el valor de una hora cambia.»

y la pregunta que hizo dirección

«¿y si la caída ocurre en temporada alta?»

→ respuesta: «entonces vale más; por eso proponemos

congelar los cambios de alto impacto en esa ventana
y ensayar la conmutación antes de que empiece»
clase 260

- se aprobó B
- y la decisión sobrevivió a dos cambios de dirección
porque el criterio estaba escrito

Decisión 2 · La que se revisó porque el supuesto cambió.

decisión original, año 0
«una sola base relacional para pedidos, inventario y
clientes»
criterio: simplicidad operativa con un equipo de 4
personas
revisión escrita: «si superamos 2.000 pedidos por minuto
en pico o si el equipo se divide en más de tres,
revisar»

lo que pasó
año 2, pico de 2.340 pedidos/minuto
→ saltó la alerta que se había puesto por esa línea

la revisión, año 2
no fue una discusión: fue abrir el registro y comprobar
el supuesto
→ 40 minutos de reunión

nueva decisión
separar inventario, que era el 71 % de la carga de
escritura
con el patrón de cuatro pasos clase 260
→ y el registro nuevo enlaza al anterior

Y lo que el equipo destacó:

la decisión original NO fue un error
fue correcta para 4 personas y 600 pedidos/minuto

→ y sin la línea de revisión, esa misma decisión se habría
discutido como si hubiera sido un error de origen
→ y quien la tomó ya no estaba en la empresa

→ eso es lo que significa que una decisión sobreviva a su
autor

Decisión 3 · La que salió mal.

qué se analizó mucho
qué biblioteca de registro usar
→ 3 semanas, comparativa de 6 opciones, 14 páginas
→ decisión reversible en 2 días

qué se decidió por costumbre
la clave de partición de la tabla de pedidos
→ «como en el proyecto anterior: por identificador de
cliente»
→ 20 minutos de conversación
→ decisión irreversible sin una migración de 7 semanas

lo que pasó
el 4 % de los clientes generaba el 61 % de los pedidos
→ particiones calientes clase 208
→ y el codo del servicio llegó mucho antes de lo
previsto clase 262

coste de la corrección, 14 meses después
migración en cuatro pasos 7 semanas
y la detención en la semana 2 por 41 divergencias por
100.000 clase 260

Y el cambio de método que salió de ahí:

se añadió un paso al principio de cada diseño
«lista las decisiones de este trabajo y marca cuáles son

caras de deshacer»
→ y el rigor se asigna a esas

efecto medido en los 18 meses siguientes

decisiones caras con registro escrito	2/9 → 11/11
decisiones baratas con análisis largo	6 → 0
tiempo total dedicado a decidir	-34 %
y decisiones caras revertidas después	3 → 0

Y el registro de decisiones como activo.

a los tres años

registros escritos	64
consultados al menos una vez	51
que provocaron una revisión por cambio de supuesto	9
y decisiones tomadas por equipos sin consultar a arquitectura, revisadas después y correctas	38 de 41

y la métrica que la persona responsable puso primero
«decisiones buenas tomadas sin mí»
→ 38 de 41

→ porque el objetivo no era decidir más: era que se decidiera bien sin cuello de botella

Y el contraste con el punto de partida:

decisiones con alternativas escritas	2/9	11/11
registros con condición de revisión	0	64
revisiones obligatorias por arquitectura	todas	solo caras
tiempo de espera por revisión	9 días	0-2 días
decisiones caras revertidas por sorpresa	3	0
diagrama que coincide con lo desplegado	no	sí

La lección que esta clase deja: la decisión sobre la biblioteca de registro consumió tres semanas y catorce páginas siendo reversible en dos días, y la clave de partición —irreversible sin una migración de siete semanas— se decidió en veinte minutos por costumbre. Y lo que hizo que una decisión de hace dos años se revisara sin drama, con su autor ya fuera de la empresa, fue una sola línea escrita: qué supuesto, si cambia, la invalida.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-22-specializations-certifications-career/272-ruta-cloud-solutions-architect/lab.py
```

El laboratorio selecciona el motor de práctica decision y produce labresult.json. El escenario, sus comprobaciones y el artefacto esperado corresponden a esta clase; no requiere credenciales y deja explícito qué debe revalidarse en un sandbox real.

1. Lee exercise.steps y formula una predicción antes de ejecutar.
2. Ejecuta la práctica y verifica todos los elementos de checks.
3. Provoca el caso de negativetest y explica la señal observada.
4. Materializa architect-plan en evidence/ usando la plantilla indicada.
5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

El artefacto principal es una matriz de decisión ponderada y un ADR. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye architect-plan para el caso CloudShop. Incluye una alternativa descartada, un supuesto que pueda falsarse, una prueba de fallo y una decisión de rollback.

■ Criterio de aceptación

- [] lab.py termina con código 0 y genera JSON válido.
- [] La entrega conecta al menos tres requisitos con mecanismos verificables.
- [] Existe una prueba positiva y una prueba negativa con evidencia.
- [] Seguridad, costo y operación aparecen como decisiones, no como anexos.
- [] Se declara una limitación y una condición que obligaría a revisar el diseño.
- [] Otra persona puede repetir el recorrido sin conocimiento tácito.

■ Errores frecuentes

Síntoma	Causa probable	Corrección
Se analizan muchas decisiones triviales y se decide rápido lo irreversible	El rigor no se asigna por reversibilidad	Al empezar cualquier diseño, lista las decisiones y marca las caras de deshacer; dedica el esfuerzo ahí y decide lo barato probando.
Una decisión antigua se discute como si hubiera sido un error	No se escribió qué supuesto la sostenía	Cierra cada registro con la condición de revisión y pon una alerta cuando el supuesto se acerque a su límite.
El diagrama no se parece a lo que está desplegado	Arquitectura de diapositivas: se diseña sin construir ni operar	Quien decide construye una parte y entra en la rotación de guardia; valida las decisiones caras con un prototipo.
Arquitectura se ha convertido en una cola de aprobaciones	Se revisan todos los diseños por sistema	Publica principios, ejemplos y registros anteriores; revisa por invitación y en las decisiones caras, y mide cuántas se toman bien sin ti.
La decisión se aprueba y luego nadie la sostiene	Se defendió la decisión en vez del criterio, o no se dijo qué empeoraba	Presenta el criterio medible, di lo que empeora antes de que lo digan, y acepta cambiar si aparece un dato nuevo.
Negocio percibe la propuesta como cara sin contexto	Se presentó el coste de lo propuesto sin el de la alternativa descartada	Lleva siempre la opción simple y el coste de lo descartado; comparar dos cifras convence mucho más que justificar una.

■ Seguridad, ética y costo

Trabaja con cuentas propias o sandboxes autorizados. No publiques secretos, identificadores reales ni datos personales. Antes de crear recursos pagos define presupuesto, etiquetas y comando de destrucción; después verifica que no queden recursos huérfanos. Los ejemplos locales enseñan contratos, pero no certifican cumplimiento ni disponibilidad de producción.

■ Preguntas de comprobación

1. ¿Qué contiene un registro de decisión y cuál es su línea más importante?
2. ¿Por qué la reversibilidad determina el rigor de una decisión?
3. ¿Qué distingue los dos modos de fracaso de esta ruta?
4. ¿Cuál es la señal de que la ruta funciona en una organización?
5. ¿Qué cuatro errores arruinan la defensa de una decisión correcta?

Referencias

- Nygard, M. (2011). Documenting architecture decisions.
<<https://cognitect.com/blog/2011/11/15/documenting-architecture-decisions>>

- Ford, N., Richards, M. y otros (2021). Software Architecture: the hard parts.
<<https://www.oreilly.com/library/view/software-architecture-the/9781492086888/>>
- Bass, L., Clements, P. y Kazman, R. (2021). Software Architecture in Practice, 4.ª ed.
<<https://www.pearson.com/en-us/subject-catalog/p/software-architecture-in-practice/P200000009360>>
- AWS (2024). Well-Architected Framework: making architectural decisions.
<<https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html>>
- Microsoft (2024). Azure architecture decision guides. <<https://learn.microsoft.com/azure/architecture/guide/>>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar:
 [Parte 22 en PDF](#)
·
 [Manual integral](#)

Anterior	Índice	Siguiente
← 271 · Ruta Cloud Data y AI Engineer	Parte 22 · Programa	273 · Mapeo AWS, Azure, Google Cloud, Kubernetes y FinOps →

Evaluación — 272 Ruta Cloud Solutions Architect

Preguntas

1. Explica el problema que resuelve esta capacidad sin mencionar un proveedor.
2. Identifica una frontera de responsabilidad y su propietario.
3. Compara dos alternativas mediante confiabilidad, seguridad, costo y operación.
4. ¿Qué demuestra el laboratorio y qué no puede demostrar?
5. Propón un caso límite o una prueba de fallo.

Reto verificable

Entrega architect-plan con diagrama o configuración, evidencia de ejecución, alternativa descartada, riesgo residual y criterio de rollback.

Criterio de aceptación

- ☐ Resultado reproducible por una segunda persona.
- ☐ Evidencia vinculada a cada afirmación técnica.
- ☐ Prueba negativa o de fallo incluida.
- ☐ Costos y permisos expresados con unidades y alcance.
- ☐ Limitaciones declaradas sin presentar la simulación como producción.

Escala

Nivel	Evidencia
A — competente	Cumple todo y defiende trade-offs con datos.
B — en desarrollo	Cumple lo esencial; falta profundidad en una dimensión.
C — inicial	Artefacto parcial o conclusión sin evidencia suficiente.
0	Sin entrega reproducible.

273 — Mapeo AWS, Azure, Google Cloud, Kubernetes y FinOps

Parte: 22 — Especializaciones, certificaciones y práctica profesional Nivel: intermedio-avanzado · Horas estimadas: 4
Laboratorio: assessment · Estado: EXECUTABLECORE

Propósito

Traducir entre proveedores sin engañarse: dar el mapa de equivalencias entre las tres nubes, los contenedores y la disciplina de coste, y —más importante— dónde la equivalencia se rompe. La clase enseña el método de traducción por restricción en vez de por nombre, y las diferencias reales que este programa ha medido y que ninguna tabla de equivalencias recoge.

Resultados de aprendizaje

Al finalizar podrás:

1. Traducir entre proveedores por restricción, no por nombre de servicio.
2. Reconocer dónde la equivalencia aparente esconde una diferencia real.
3. Usar el mapa de las cinco capas para orientarse en una nube nueva.
4. Evaluar una traducción con las preguntas que la validan.
5. Aplicar el mismo método a los contenedores y a la disciplina de coste.

Conceptos centrales

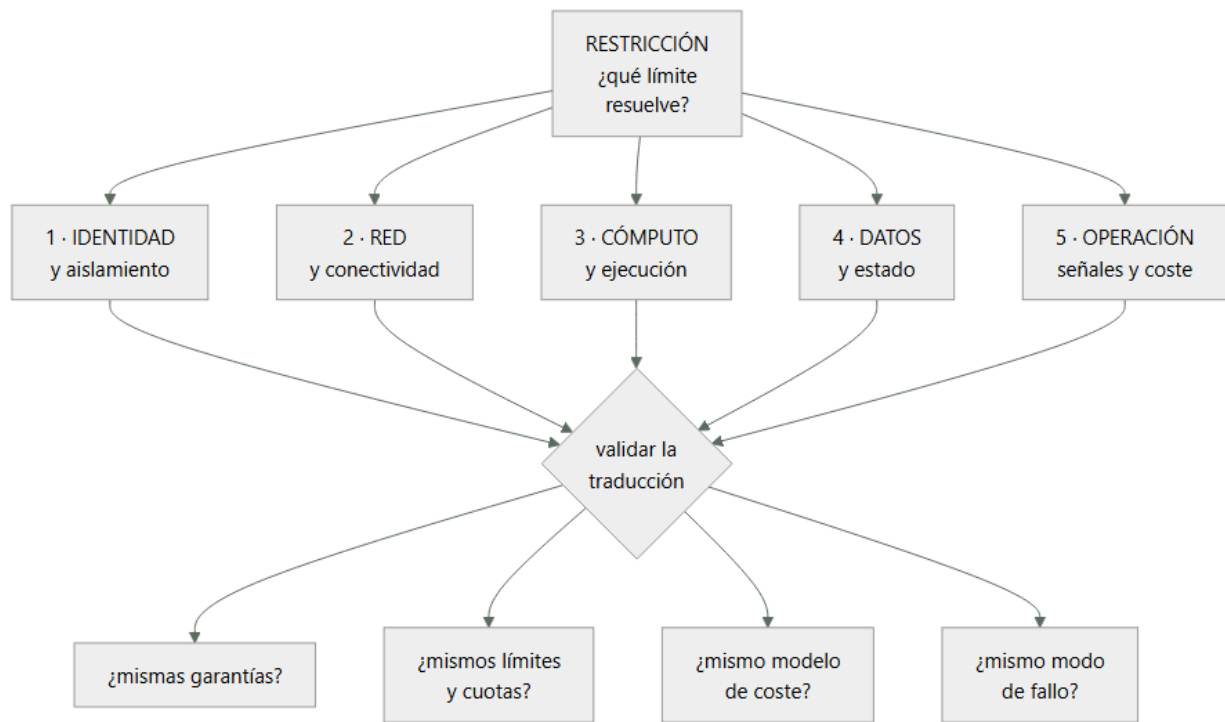
Concepto	Comprensión verificable
traducción por restricción	Buscar cómo la otra nube resuelve el mismo límite físico, en vez de buscar el servicio con nombre parecido.
equivalencia falsa	Dos servicios con el mismo propósito y garantías, límites o modelo de coste distintos.
unidad de aislamiento	El contenedor administrativo donde vive la separación: cuenta, suscripción o proyecto.
plano de control	La interfaz que crea y modifica recursos. Sus cuotas y su latencia difieren mucho entre nubes.
portabilidad aparente	Crear que usar la misma tecnología en tres nubes elimina las diferencias. No lo hace.
coste de traducción	Lo que cuesta operar en dos nubes: no es el doble de aprender, es el doble de operar.

Modelo mental

Una especialización combina fundamentos, evidencia de proyectos y juicio bajo restricciones; una insignia sin práctica no sustituye esa combinación.

Aplicado a esta clase, separa siempre cuatro planos: intención (qué necesita el usuario), configuración (qué declaramos), estado observado (qué existe de verdad) y evidencia (cómo sabemos que cumple). Confundirlos produce diseños que se ven correctos en un diagrama pero fallan al operar.

■ Flujo de razonamiento



Desarrollo

1. Traducir por restricción, no por nombre

Las tablas de equivalencias por nombre producen errores caros. El método que funciona empieza por la restricción.

EL MÉTODO

- 1 ¿qué RESTRICCIÓN resuelve esto?
→ «necesito que un servicio pruebe su identidad ante otro sin credencial permanente»
- 2 ¿cómo la resuelve la nube destino?
- 3 ¿con qué garantías, límites y coste?
- 4 ¿y con qué modo de fallo?

→ y solo entonces se escribe el nombre del servicio
→ empezar por el nombre es lo que produce las sorpresas

Y las cinco capas en las que conviene organizarse:

- 1 IDENTIDAD Y AISLAMIENTO
quién es quién, qué puede, y dónde vive la separación
→ cuenta · suscripción · proyecto
- 2 RED Y CONECTIVIDAD
direccionamiento, rutas, resolución de nombres, salida a internet, interconexión
- 3 CÓMPUTO Y EJECUCIÓN
máquinas, contenedores, funciones, escalado
- 4 DATOS Y ESTADO
objetos, bloques, relacional, no relacional, analítico, mensajería
- 5 OPERACIÓN
señales, registros, políticas, coste, cuotas

→ en toda nube existen las cinco
→ y aprender una nube nueva es rellenar esta rejilla, no leer un catálogo

Y el mapa por restricción, que es lo transferible:

RESTRICCIÓN	DÓNDE MIRAR EN CUALQUIER NUBE
«que un servicio pruebe quién es sin credencial fija»	identidad de carga de trabajo, federada
«separar entornos para que un fallo no cruce»	unidad de aislamiento administrativa

«que el tráfico solo llegue a quien debe»	segmentación de red y reglas por identidad
«que un cambio no afecte a todos a la vez»	despliegue progresivo del proveedor o propio
«que el dato sobreviva a un administrador comprometido»	copia en otra unidad de aislamiento, inmutable
«saber quién gasta qué»	etiquetas y jerarquía administrativa
«no pasar de N peticiones por segundo al plano de control»	cuotas del plano de control

→ y este mapa no cambia cuando cambian los productos

2. Dónde la equivalencia se rompe

Lo que ninguna tabla de equivalencias dice, y este programa ha medido.

- 1 LA UNIDAD DE AISLAMIENTO NO ES EQUIVALENTE
cuenta, suscripción y proyecto se parecen y difieren en cuánto cuesta crear una nueva
qué se hereda desde arriba
y qué queda compartido aunque parezca separado
→ y de aquí salen los errores de aislamiento más caros
clases 219, 231
- 2 LOS LÍMITES POR DEFECTO SON MUY DISTINTOS
dos nubes con el «mismo» servicio de funciones tienen concurrencias por defecto de órdenes distintos
→ y en la región secundaria, los valores por defecto mandan
clase 262, ley 26
- 3 EL MODELO DE COSTE CAMBIA LA ARQUITECTURA
donde el tráfico entre zonas se cobra, la arquitectura correcta es otra
donde el análisis se cobra por dato leído, el formato y la partición valen más que el motor clase 243, ley 28
→ y una traducción literal puede multiplicar la factura
- 4 LA COHERENCIA DE LOS ALMACENES DE OBJETOS
parecen iguales y sus garantías de lectura tras escritura, de listado y de sobrescritura no lo son
→ y esto rompe canales de datos silenciosamente
ley 29
- 5 LA RESOLUCIÓN DE NOMBRES Y LA SALIDA A INTERNET
quién resuelve, desde dónde, y qué pasa con las respuestas privadas difiere mucho clase 195
→ es la fuente número uno de «funciona en una nube y no en la otra»
- 6 EL PLANO DE CONTROL
su latencia y sus cuotas de peticiones difieren
→ y una herramienta de infraestructura como código que funciona en una nube se estrangula en otra
clases 128, 262
- 7 Y LA SEMÁNTICA DE LOS PERMISOS
denegar explícito, herencia, ámbito y evaluación no funcionan igual
→ una política traducida literalmente puede conceder más de lo que concedía clase 231

Y la regla que resume todo lo anterior:

LO QUE SE TRADUCE BIEN
el propósito y la forma de la arquitectura

LO QUE NO SE TRADUCE
garantías, límites, modelo de coste y modos de fallo

→ y esos cuatro son los que deciden si el sistema funciona

→ por eso la traducción se VALIDA midiendo, no leyendo

3. Contenedores y la portabilidad aparente

El caso especial que más expectativas defrauda.

LO QUE SÍ APORTA UNA CAPA COMÚN DE CONTENEDORES
el empaquetado y el modelo de despliegue se parecen
las descripciones de carga de trabajo son casi iguales
y el conocimiento del equipo se traslada

LO QUE NO ELIMINA
la identidad y su federación con la nube
la red: direccionamiento, salida, balanceo, resolución
el almacenamiento persistente y sus garantías
el registro de imágenes y su disponibilidad
las cuotas del plano de control
el coste, muy distinto por nube
y la operación: actualizaciones, versiones y su ritmo

→ la portabilidad es del ARTEFACTO, no del SISTEMA

→ y la parte que no se traslada es la que da los
incidentes clase 185

Y el coste real de operar en dos nubes:

NO ES EL DOBLE DE APRENDER: ES EL DOBLE DE OPERAR
dos inventarios clase 253
dos modelos de permisos clase 231
dos formas de red clase 194
dos conjuntos de cuotas clase 262
dos catálogos de alertas y dos guardias clase 257
dos formas de facturar clase 270
y ensayos en las dos clase 261

→ y por eso multinube por defecto es caro

→ y multinube por una RAZÓN concreta –normativa,
adquisición, un servicio que solo existe en una– se
sostiene

y la pregunta que ordena la decisión

«¿qué riesgo concreto estamos comprando con este coste?»

→ si la respuesta es «no depender de un proveedor», hay
que cuantificar cuánto vale eso y compararlo

Y el mapa de la disciplina de coste, que también se traduce:

lo que existe en las tres, con otros nombres
jerarquía administrativa para atribuir
etiquetas y su obligatoriedad
exportación detallada de facturación
compromisos con descuento
presupuestos y alertas
y capacidad interrumpible

lo que cambia de verdad
qué se cobra por transferencia entre zonas y regiones
cómo se factura el almacenamiento frío y su
recuperación
cómo se cobra la analítica: por dato leído o por
capacidad reservada
y el retraso con que llega el detalle de coste

→ y esos cuatro cambian decisiones de diseño
clase 270

4. Cómo se valida una traducción

Una traducción sobre el papel no vale nada. Estas son las comprobaciones que la convierten en fiable.

LAS CUATRO PREGUNTAS DE VALIDACIÓN

- 1 ¿MISMAS GARANTÍAS?
coherencia, durabilidad, orden, exactamente una vez
→ y aquí casi siempre hay una diferencia
- 2 ¿MISMOS LÍMITES Y CUOTAS?
por defecto y ampliables, por región
- 3 ¿MISMO MODELO DE COSTE?

qué se cobra por operación, por dato, por hora
4 ¿MISMO MODO DE FALLO?
¿falla rápido o se degrada? ¿qué se ve cuando falla?

→ y las cuatro se responden con documentación y con una prueba

Y la prueba mínima antes de comprometerse:

PROTOTIPO DE UNA SEMANA

el camino crítico de extremo a extremo
con identidad real, red real y datos de tamaño realista
midiendo latencia, coste por operación y comportamiento al fallar
y provocando un fallo a propósito clase 261

→ una semana de esto ahorra meses
→ y descubre las diferencias que ninguna tabla recoge

Y cómo se demuestra esta competencia:

LO QUE NO VALE

«conozco las tres nubes»
→ y la parte 22 predijo que esto rendiría peor que conocer una a fondo clase 264

LO QUE VALE

«migramos el canal de datos y descubrimos que las garantías de listado del almacén de objetos eran distintas; lo detectamos con una prueba de una semana antes de comprometernos»
«traducimos la política y concedía más de lo que concedía la original; lo cogió la comprobación automática»
«el mismo diseño costaba 3,2 veces más por el cobro de tráfico entre zonas, y rediseñamos la colocación»

→ efecto, mecanismo y cifra clase 275

Y la lista de comprobación de la clase:

- traduzco por restricción, no por nombre de servicio
- uso la rejilla de cinco capas para orientarme
- compruebo garantías, límites, coste y modo de fallo
- sé que la unidad de aislamiento no es equivalente
- reviso los valores por defecto, sobre todo en la región secundaria
- compruebo la semántica de permisos, no solo su forma
- verifico resolución de nombres y salida a internet
- vigilo las cuotas del plano de control
- no confundo portabilidad del artefacto con la del sistema
- cuantifico el coste de operar en dos nubes
- hago un prototipo de una semana antes de comprometerme
- y provoqué un fallo en ese prototipo

Y el cierre que enlaza con la clase siguiente: con el mapa y el método de traducción, queda la parte que examina esto por escrito y con tiempo limitado. Las preguntas de escenario y la estrategia de examen son la materia de la clase 274.

Ejemplo trabajado

CloudShop traduce tres piezas entre nubes. Lo que sigue son las cuatro diferencias que el prototipo de una semana encontró y ninguna tabla recogía, la política traducida que concedía de más, y la cuenta del coste real de operar en dos nubes.

Traducción 1 · El canal de datos.

qué se traducía
ingesta a almacén de objetos, catálogo, y consultas analíticas clase 242

la tabla de equivalencias decía
almacén de objetos → almacén de objetos

catálogo → catálogo
motor de consulta → motor de consulta

→ traducción aparentemente trivial

Y lo que el prototipo de una semana encontró:

DIFERENCIA 1 · garantías de listado
el proceso dependía de listar un prefijo justo después
de escribir
→ en el origen, el listado era inmediato
→ en el destino, podía tardar
→ y el trabajo posterior procesaba ficheros de menos
SIN ERROR ley 29

cómo se detectó
prueba con 40.000 ficheros escritos y listados de
inmediato
→ 61 casos de listado incompleto
cómo se arregló
manifiesto explícito de ficheros, en vez de listar

DIFERENCIA 2 · modelo de coste de la consulta
origen por dato leído
destino por capacidad reservada
→ el mismo conjunto de 41 informes
origen, tras optimizar 12 USD/informe
destino con capacidad mínima por debajo del
umbral, pero con
suelo mensual
→ por debajo de cierto volumen el destino era más caro
→ y por encima, mucho más barato
→ la decisión dependía del volumen, no del motor

DIFERENCIA 3 · cuota del plano de control
la herramienta de infraestructura como código aplicaba
341 recursos
→ en el destino se estrangulaba a mitad
→ aplicación fallida de forma intermitente clase 128
→ se resolvió partiendo el estado y limitando el
paralelismo

DIFERENCIA 4 · zonas horarias en las particiones
la convención por defecto del catálogo difería
→ y las particiones diarias se desplazaban unas horas
→ los informes diarios cuadraban salvo en el límite del
día clase 243

Y la valoración:

coste del prototipo	1 semana, 1 persona
diferencias encontradas	4
que ninguna tabla de equivalencias recogía	4
que habrían llegado a producción sin error visible	2

→ las dos silenciosas eran las peligrosas

Traducción 2 · La política que concedía de más.

política original
permitir lectura de un prefijo concreto del almacén, a
un rol de servicio, con denegación explícita para todo
lo demás clase 231

la traducción literal
se mantuvo la concesión y se omitió la denegación
explícita
→ porque en la nube destino la herencia y la evaluación
funcionaban distinto y «parecía innecesaria»

lo que produjo
el rol heredaba desde el nivel superior un permiso de
lectura sobre el contenedor entero administrativo
→ y sin la denegación explícita, la herencia ganaba
→ el rol podía leer 214 conjuntos de datos en vez de 1

Y cómo se detectó:

no en la revisión humana: la revisión lo aprobó

lo cogió la comprobación automática de permisos efectivos
→ que compara «qué puede hacer realmente esta identidad»
antes y después clase 217
→ 214 recursos accesibles frente a 1 esperado

→ la lección: se compara el EFECTO, no el texto de la política
→ y esa comprobación se añadió a la cadena para toda traducción

La cuenta del coste real de operar en dos nubes.

CloudShop tenía una razón concreta para la segunda nube
un cliente del sector público exigía residencia en una región que el proveedor principal no ofrecía
clase 280

y se midió lo que costaba mantenerla

inventario y etiquetado, segunda nube	0,4 personas
modelo de permisos y su auditoría	0,3
red y conectividad	0,3
cuotas y capacidad	0,2
alertas, guardia y procedimientos	0,6
facturación y atribución	0,2
ensayos	0,2
actualizaciones y versiones	0,3
	■■■■■■■
	2,5 personas
más el coste de infraestructura	41.000 USD/mes
y el sobre coste por menor volumen (compromisos peores)	+9 %

Y la decisión que se tomó con esa cifra:

ingresos del cliente que exigía la segunda nube
1,9 M USD/año
coste de mantenerla ~2,5 personas + 492.000 USD/año

→ se mantuvo, y con la cifra escrita en el registro de decisión clase 272
→ con condición de revisión: «si estos ingresos bajan de 900.000, se revisa»

y lo que NO se hizo
no se replicó el resto de la plataforma en la segunda nube «por si acaso»
→ solo lo que ese cliente necesitaba
→ 6 servicios de 41

Y la comparación con lo que se había propuesto al principio:

propuesta inicial	«seamos multinube en todo, para no depender de un proveedor»
coste estimado	~7 personas + 1,4 M USD/año
riesgo evitado	no cuantificado

y la pregunta que la desmontó
«¿cuál es el escenario concreto del que nos protege, y cuánto nos costaría si ocurriera?»
→ el escenario realista era una subida de precios o una pérdida de servicio prolongada
→ y la protección real requería mantener el sistema EJECUTÁNDOSE en las dos, no solo poder migrar
→ lo que multiplicaba la cifra

→ se decidió: multinube donde hay una razón concreta; portabilidad razonable en lo demás
→ y «portabilidad razonable» se definió: contratos, datos

exportables y decisiones registradas, no abstracciones
propias clase 267

Y la rejilla de cinco capas, rellena para orientarse.

el equipo mantuvo una tabla de una página por nube

capa	qué mirar	dónde difiere más
identidad	unidad de aislamiento y federación	herencia y denegación
red	direccionamiento, salida, balanceo	resolución de nombres privados
cómputo	escalado, arranque, interrumpible	límites por defecto
datos	garantías del almacén y del relacional	listado y coste por operación
operación	señales, políticas, cuotas, coste	retraso del detalle de coste

→ una página por nube, actualizada cuando algo sorprende
→ y usada para entrar en una nube nueva en días

La lección que esta clase deja: la traducción del canal de datos parecía trivial y el prototipo de una semana encontró cuatro diferencias, dos de ellas silenciosas —un listado incompleto y un desplazamiento de particiones— que habrían llegado a producción sin producir ningún error. Y la política traducida pasó la revisión humana mientras concedía acceso a 214 conjuntos de datos en vez de uno: lo cogió comparar permisos efectivos, no leer el texto.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-22-specializations-certifications-career/273-mapeo-aws-azure-google-cloud-  
kubernetes-y-finops/lab.py
```

El laboratorio selecciona el motor de práctica assessment y produce labresult.json. El escenario, sus comprobaciones y el artefacto esperado corresponden a esta clase; no requiere credenciales y deja explícito qué debe revalidarse en un sandbox real.

1. Lee exercise.steps y formula una predicción antes de ejecutar.
2. Ejecuta la práctica y verifica todos los elementos de checks.
3. Provoca el caso de negativetest y explica la señal observada.
4. Materializa certification-map en evidence/ usando la plantilla indicada.
5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

El artefacto principal es una evaluación por escenarios con rúbrica y evidencia trazable. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye certification-map para el caso CloudShop. Incluye una alternativa descartada, un supuesto que pueda falsarse, una prueba de fallo y una decisión de rollback.

■ Criterio de aceptación

- [] lab.py termina con código 0 y genera JSON válido.
- [] La entrega conecta al menos tres requisitos con mecanismos verificables.
- [] Existe una prueba positiva y una prueba negativa con evidencia.
- [] Seguridad, costo y operación aparecen como decisiones, no como anexos.
- [] Se declara una limitación y una condición que obligaría a revisar el diseño.
- [] Otra persona puede repetir el recorrido sin conocimiento tácito.

■ Errores frecuentes

Síntoma	Causa probable	Corrección
Un servicio equivalente se comporta distinto en producción	Se tradujo por nombre y no se compararon garantías, límites, coste y modo de fallo	Traduce por restricción y valida las cuatro preguntas con documentación y con un prototipo del camino crítico.
Una política traducida concede más de lo que concedía	La herencia y la evaluación de permisos difieren y la denegación explícita se omitió	Compara permisos efectivos antes y después, no el texto de la política; automatiza esa comprobación en la cadena.
El canal de datos procesa registros de menos sin dar error	Las garantías de listado y de lectura tras escritura del almacén no son iguales	Usa manifiestos explícitos en vez de listar, y prueba con volumen realista escribiendo y listando de inmediato.
La infraestructura como código se aplica a medias e intermitentemente	Se alcanzan las cuotas de peticiones del plano de control, que difieren por nube	Parte el estado, limita el paralelismo y vigila la cuota del plano de control como una más del inventario.
El mismo diseño cuesta varias veces más en la otra nube	El modelo de coste cambia lo que es una buena arquitectura	Revisa qué se cobra por transferencia, por operación y por dato leído; rediseña colocación y formato antes de migrar.
Se adopta multinube por no depender de un proveedor y el coste se dispara	No se cuantificó el riesgo evitado ni el coste de operar dos nubes	Exige un escenario concreto y su coste; multinube donde hay una razón, portabilidad razonable en lo demás.

■ Seguridad, ética y costo

Trabaja con cuentas propias o sandboxes autorizados. No publiques secretos, identificadores reales ni datos personales. Antes de crear recursos pagos define presupuesto, etiquetas y comando de destrucción; después verifica que no queden recursos huérfanos. Los ejemplos locales enseñan contratos, pero no certifican cumplimiento ni disponibilidad de producción.

■ Preguntas de comprobación

1. ¿Por qué se traduce por restricción y no por nombre de servicio?
2. ¿Cuáles son las cinco capas de la rejilla y para qué sirven?
3. ¿Qué cuatro cosas no se traducen nunca y por qué deciden el resultado?
4. ¿Qué aporta y qué no aporta una capa común de contenedores?
5. ¿Qué comprueba un prototipo de una semana que ninguna tabla recoge?

Referencias

- AWS (2024). Compare AWS and Azure services. <<https://learn.microsoft.com/azure/architecture/aws-professional/services>>
- Google Cloud (2024). Google Cloud for AWS professionals. <<https://cloud.google.com/docs/get-started/aws-comparison>>
- CNCF (2024). Kubernetes conformance and portability. <<https://www.cncf.io/certification/software-conformance/>>
- FinOps Foundation (2024). FOCUS: FinOps cost and usage specification — coste comparable entre nubes. <<https://focus.finops.org/>>
- Brewer, E. (2012). CAP twelve years later: how the rules have changed — por qué las garantías no se traducen. <<https://www.infoq.com/articles/cap-twelve-years-later-how-the-rules-have-changed/>>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 22 en PDF · Manual integral

Anterior	Índice	Siguiente
← 272 · Ruta Cloud Solutions Architect	Parte 22 · Programa	274 · Preguntas de escenario y estrategia de examen →

Evaluación — 273 Mapeo AWS, Azure, Google Cloud, Kubernetes y FinOps

Preguntas

1. Explica el problema que resuelve esta capacidad sin mencionar un proveedor.
2. Identifica una frontera de responsabilidad y su propietario.
3. Compara dos alternativas mediante confiabilidad, seguridad, costo y operación.
4. ¿Qué demuestra el laboratorio y qué no puede demostrar?
5. Propón un caso límite o una prueba de fallo.

Reto verificable

Entrega certification-map con diagrama o configuración, evidencia de ejecución, alternativa descartada, riesgo residual y criterio de rollback.

Criterio de aceptación

- ☐ Resultado reproducible por una segunda persona.
- ☐ Evidencia vinculada a cada afirmación técnica.
- ☐ Prueba negativa o de fallo incluida.
- ☐ Costos y permisos expresados con unidades y alcance.
- ☐ Limitaciones declaradas sin presentar la simulación como producción.

Escala

Nivel	Evidencia
A — competente	Cumple todo y defiende trade-offs con datos.
B — en desarrollo	Cumple lo esencial; falta profundidad en una dimensión.
C — inicial	Artefacto parcial o conclusión sin evidencia suficiente.
0	Sin entrega reproducible.

274 — Preguntas de escenario y estrategia de examen

Parte: 22 — Especializaciones, certificaciones y práctica profesional Nivel: intermedio-avanzado · Horas estimadas: 4
Laboratorio: assessment · Estado: EXECUTABLECORE

Propósito

Resolver preguntas de escenario y aprobar exámenes de certificación sin confundirlos con capacidad. La clase da el método para leer un escenario —encontrar la restricción que decide, descartar por eliminación, reconocer las señales de la respuesta esperada—, la estrategia de tiempo, y lo que un examen no puede medir y por tanto hay que demostrar en otro sitio.

Resultados de aprendizaje

Al finalizar podrás:

1. Leer un escenario identificando la restricción que decide.
2. Descartar opciones por eliminación con criterios explícitos.
3. Reconocer las señales que delatan la respuesta esperada.
4. Gestionar el tiempo y las preguntas dudosas.
5. Situar la certificación en su valor real, sin sobreestimarla ni despreciarla.

Conceptos centrales

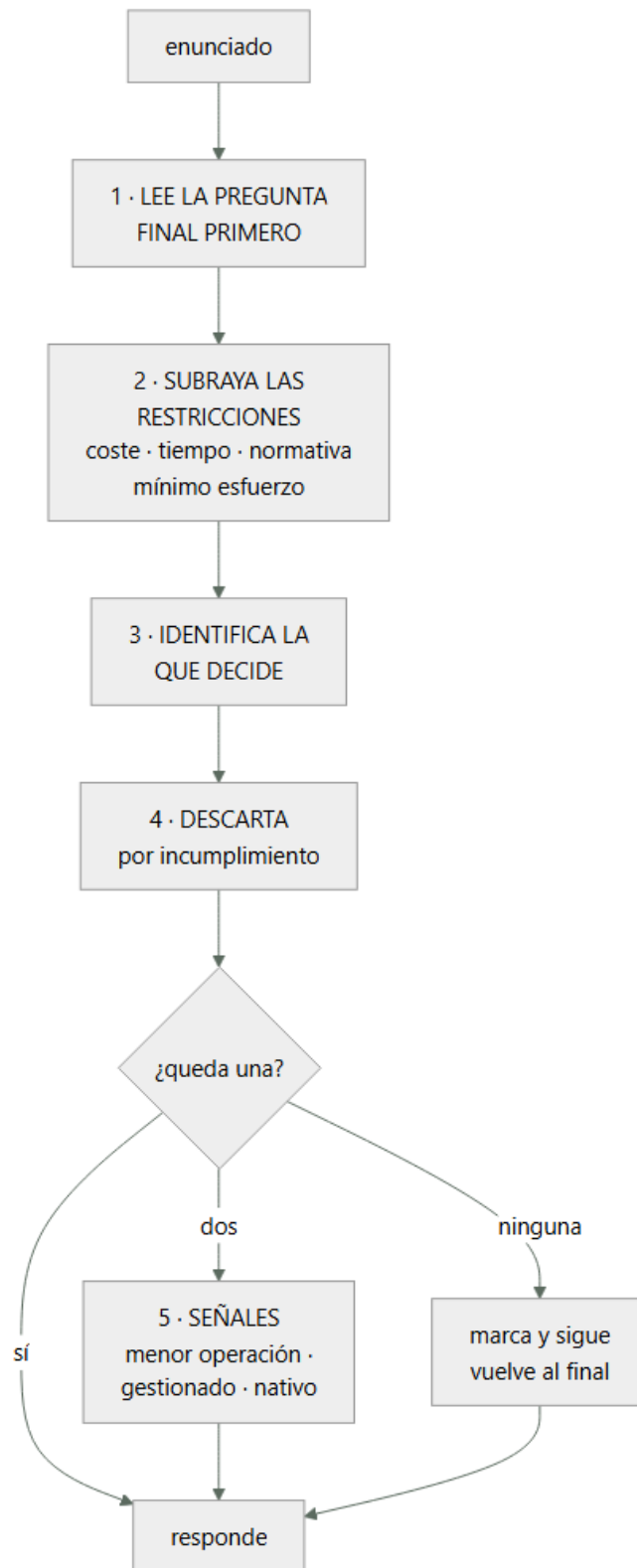
Concepto	Comprensión verificable
pregunta de escenario	Enunciado con contexto y varias opciones plausibles, donde una restricción decide cuál es correcta.
restricción que decide	La condición del enunciado que elimina todas las opciones menos una. Suele ir en una sola frase.
distractor	Opción técnicamente posible pero que incumple una restricción del enunciado.
respuesta esperada	La que el examen considera correcta según la doctrina del proveedor, que puede no ser la mejor en tu contexto.
coste de oportunidad del examen	Lo que se deja de aprender mientras se estudia para aprobar.
conocimiento inerte	El que se recuerda para el examen y no se aplica después.

Modelo mental

Una especialización combina fundamentos, evidencia de proyectos y juicio bajo restricciones; una insignia sin práctica no sustituye esa combinación.

Aplicado a esta clase, separa siempre cuatro planos: intención (qué necesita el usuario), configuración (qué declaramos), estado observado (qué existe de verdad) y evidencia (cómo sabemos que cumple). Confundirlos produce diseños que se ven correctos en un diagrama pero fallan al operar.

■ Flujo de razonamiento



Desarrollo

1. Cómo se lee un escenario

Las preguntas de escenario están construidas con un patrón, y conocerlo cambia mucho el resultado.

LA ESTRUCTURA HABITUAL

contexto 2-6 líneas, con datos relevantes e irrelevantes mezclados

requisito	lo que hay que conseguir
restricciones	coste mínimo · mínimo esfuerzo operativo · sin interrumpir el servicio · en menos de X · sin cambiar la aplicación
pregunta final	qué hay que elegir
opciones	una correcta y tres plausibles

- y las restricciones son lo que decide
- casi nunca decide la parte técnica del contexto

Y el método, en orden:

- 1 LEE LA PREGUNTA FINAL ANTES QUE EL CONTEXTO
 - así lees el contexto buscando algo concreto
 - y no memorizas datos irrelevantes
- 2 SUBRAYA LAS RESTRICCIONES
 - «con el menor coste posible»
 - «con el menor esfuerzo operativo»
 - «sin modificar la aplicación»
 - «sin tiempo de inactividad»
 - «cumpliendo la normativa X»
 - «lo antes posible»
- 3 IDENTIFICA LA QUE DECIDE
 - normalmente una sola elimina tres opciones
- 4 DESCARTA POR INCUMPLIMIENTO, no por preferencia
 - «esta requiere cambiar la aplicación» → fuera
 - «esta implica una ventana de parada» → fuera
- 5 SI QUEDAN DOS, aplica las señales

Y las restricciones más frecuentes y lo que implican:

- «MENOR ESFUERZO OPERATIVO»
 - prefiere el servicio gestionado sobre el autogestionado
 - prefiere lo nativo sobre lo montado a mano
 - elimina cualquier opción con máquinas que mantener
- «MENOR COSTE»
 - prefiere almacenamiento frío, capacidad interrumpible, procesamiento por lotes
 - cuidado: a veces la más barata incumple otra restricción
- «SIN CAMBIAR LA APLICACIÓN»
 - elimina toda opción que exija tocar el código
 - suele apuntar a una capa intermedia o a un intermediario
- «SIN INTERRUPCIÓN»
 - elimina migraciones con parada
 - apunta a replicar y conmutar, o a pasos reversibles
- «LO ANTES POSIBLE»
 - prefiere lo que ya existe sobre lo que hay que construir

2. Las señales y los distractores

Cuando dos opciones sobreviven al descarte, hay patrones que resuelven la mayoría.

- SEÑALES DE LA RESPUESTA ESPERADA
- usa el servicio gestionado del proveedor antes que uno montado
 - resuelve dentro de la nube antes que con una pieza externa
 - usa identidades y roles antes que claves
 - usa cifrado gestionado antes que propio
 - usa la opción con menos piezas
 - y respeta el privilegio mínimo

ley 26

- la doctrina del proveedor es coherente: gestionado, nativo, con identidad, con menos operación
- y las preguntas la reflejan

Y los distractores típicos, que se reconocen a simple vista:

- 1 EL QUE HACE DEMASIADO
una arquitectura elaborada para un problema sencillo
→ si el enunciado pide algo simple, la respuesta es simple
- 2 EL QUE USA CREDENCIALES PERMANENTES
«crea una clave de acceso y guárdala en el código»
→ casi siempre incorrecto clase 256
- 3 EL QUE CONCEDE PERMISOS AMPLIOS
«asigna el rol de administrador»
→ casi siempre incorrecto clase 231
- 4 EL QUE INTRODUCE UNA MÁQUINA QUE MANTENER
cuando la restricción dice «mínimo esfuerzo operativo»
- 5 EL TÉCNICAMENTE CORRECTO QUE INCUMPLE UNA RESTRICCIÓN
→ el más peligroso: funciona, pero el enunciado lo prohibía
→ por eso se subrayan las restricciones primero
- 6 Y EL QUE INVENTA UNA CAPACIDAD
un servicio que no hace lo que la opción dice
→ y este se reconoce si se conoce el servicio de verdad

Y las palabras del enunciado que suelen ser trampa:

«actualmente»	describe el problema, no la solución
«el equipo no tiene experiencia en»	elimina las opciones que la requieren
«debe mantenerse durante 7 años»	apunta a archivo y a inmutabilidad clase 255
«los datos no pueden salir de la región»	apunta a residencia y elimina servicios globales
«el pico ocurre una vez al mes»	apunta a elasticidad, no a capacidad reservada
y «se accede pocas veces al año»	apunta a almacenamiento frío y a su coste de recuperación

3. Estrategia de tiempo y de dudas

La gestión del examen es una habilidad separada del conocimiento, y explica una parte de los suspensos.

- EL PRESUPUESTO DE TIEMPO**
divide el tiempo total entre las preguntas y ten el ritmo objetivo en la cabeza
→ si vas por detrás a mitad, acelera antes de que sea tarde

- LAS TRES PASADAS**
- 1ª responde lo que sabes; marca lo dudoso; no te atasques
→ nunca dejes una sin responder: no hay penalización por fallar
 - 2ª vuelve a lo marcado, con tiempo conocido
 - 3ª si queda tiempo, revisa solo lo marcado como muy dudoso

- y NO revises lo que respondiste con seguridad
- cambiar respuestas seguras empeora el resultado más veces de las que lo mejora

Y qué hacer con una pregunta que no se sabe:

- 1 descarta lo que incumple una restricción
- 2 descarta credenciales permanentes y permisos amplios
- 3 descarta lo que hace demasiado
- 4 y entre lo que quede, elige lo gestionado y con menos piezas

→ esto convierte una adivinanza de 1 entre 4 en una de 1 entre 2, con frecuencia

Y los errores de examen más comunes:

leer el contexto entero y luego la pregunta
 → se pierde tiempo y se retiene lo irrelevante
 elegir la que uno haría en su trabajo
 → el examen pregunta la doctrina del proveedor
 → y a veces tu solución es mejor y no es la esperada
 no leer «elige DOS» o «elige TRES»
 quedarse en una pregunta más de dos minutos
 y cambiar respuestas por nerviosismo al final

Y cómo se prepara de verdad:

QUÉ FUNCIONA

hacer preguntas de práctica y ANALIZAR LOS FALLOS
 → por qué la correcta era correcta y por qué la mía no
 construir lo que se estudia, aunque sea pequeño
 y leer la documentación de los límites y las cuotas
 → que es lo que más se pregunta y menos se estudia

QUÉ NO FUNCIONA

memorizar volcados de preguntas
 → además de inútil, suele vulnerar las condiciones del examen
 ver videos sin construir nada
 y estudiar servicios que el examen apenas toca

4. Qué mide y qué no mide un examen

La parte honesta, que conviene tener clara antes de invertir semanas.

LO QUE UN EXAMEN SÍ MIDE

amplitud de catálogo: qué servicio sirve para qué
 conocimiento de límites, cuotas y modelos de precio
 la doctrina del proveedor
 y capacidad de descartar bajo restricciones

→ y esto tiene valor real: acelera conversaciones y evita reinventar

LO QUE NO PUEDE MEDIR

decidir con información incompleta	
diagnosticar bajo presión	clase 258
sostener un número ante negocio	clase 268
saber qué NO automatizar	clase 259
y el juicio que no está escrito	ley 30

→ porque nada de eso se puede examinar por opción múltiple

→ y es justo donde está la diferencia entre nivel 2 y nivel 3 clase 265

Y para qué sirve una certificación, sin exagerar:

SIRVE PARA

pasar filtros de contratación
 dar estructura a un estudio disperso
 cubrir huecos de catálogo que uno no sabía que tenía
 y requisitos de socio comercial de la empresa

NO SIRVE PARA

demostrar que sabes resolver	
sustituir evidencia de trabajo real	clase 275
ni justificar un nivel	

→ y el coste de oportunidad es real: seis semanas de

estudio son seis semanas de no construir
→ la combinación que funciona es certificación + evidencia
→ la certificación abre la puerta y la evidencia
sostiene la conversación

Y el riesgo del conocimiento inerte:

SE APRUEBA Y A LOS SEIS MESES NO SE RECUERDA NADA
porque no se aplicó

LA DEFENSA

construir algo con lo que se estudió, aunque sea pequeño
→ y escribirlo, que es lo que lo convierte en evidencia
clase 275
y elegir el examen que se solapa con lo que haces
ahora

Y la lista de comprobación de la clase:

- leo la pregunta final antes que el contexto
- subrayo las restricciones y encuentro la que decide
- descarto por incumplimiento, no por preferencia
- reconozco los distractores de credenciales y permisos
- respondo todas: no hay penalización por fallar
- hago tres pasadas y no reviso lo que sabía
- no me quedo más de dos minutos en una pregunta
- analizo por qué fallé cada pregunta de práctica
- estudio límites, cuotas y precios, que es lo que más cae
- construyo algo con lo que estudio
- y sé qué mide el examen y qué tengo que demostrar en otro sitio

Y el cierre que enlaza con la clase siguiente: el examen abre la puerta y no sostiene la conversación. Lo que la sostiene es la evidencia de trabajo real, bien presentada. Portafolio, evidencia y entrevista de sistemas es la materia de la clase 275.

Ejemplo trabajado

Cinco preguntas de escenario resueltas con el método, y los datos de preparación de cuatro personas del equipo de CloudShop. Lo que sigue muestra dónde estaba la restricción que decidía en cada caso.

Pregunta 1 · La restricción está en una sola palabra.

ENUNCIADO

«Una empresa almacena registros de auditoría que deben conservarse 7 años por normativa. Se consultan menos de dos veces al año, y cuando se consultan puede tardarse varias horas en recuperarlos. ¿Qué opción cumple con el menor coste?»

- A almacenamiento estándar con ciclo de vida a 90 días
- B almacenamiento de archivo profundo con bloqueo de retención
- C base de datos gestionada con copias mensuales
- D almacenamiento estándar con replicación entre regiones

MÉTODO

restricciones subrayadas
7 años por normativa · menos de 2 accesos al año ·
puede tardar varias horas · menor coste

la que decide: «puede tardar varias horas»
→ autoriza el archivo profundo, que es lo más barato
→ si no lo dijera, el archivo profundo quedaría descartado

descarte

- A no cumple 7 años
- C base de datos para registros de auditoría: caro y no aporta
- D replicación no la pide nadie: coste innecesario

RESPUESTA B

y el bloqueo de retención está ahí por «normativa»
clase 255

Pregunta 2 · El distractor técnicamente correcto.

ENUNCIADO

«Una aplicación que se ejecuta en máquinas virtuales necesita leer objetos de un contenedor de almacenamiento. La empresa exige que no se almacenen credenciales de larga duración. ¿Qué haces?»

- A crear una clave de acceso y guardarla cifrada en el gestor de secretos
- B asignar un rol a la instancia con permiso de solo lectura sobre ese contenedor
- C asignar un rol a la instancia con permiso de administrador de almacenamiento
- D usar credenciales de un usuario de servicio rotadas cada 30 días

MÉTODO

restricción que decide: «no credenciales de larga duración»

- A técnicamente funciona y guarda una credencial de larga duración → fuera
- D igual: rotar no deja de ser una credencial permanente → fuera
- C cumple la restricción y viola privilegio mínimo

RESPUESTA B

→ A y D son el distractor más común: funcionan, y el enunciado los prohibía

Pregunta 3 · «Sin cambiar la aplicación».

ENUNCIADO

«Una aplicación heredada abre demasiadas conexiones a la base de datos y la satura en picos. No se puede modificar el código. ¿Qué solución aplicas con el menor esfuerzo operativo?»

- A reescribir la capa de acceso para usar un grupo de conexiones
- B desplegar un intermediario de conexiones gestionado entre la aplicación y la base
- C aumentar el tamaño de la instancia de base de datos
- D instalar un intermediario en una máquina propia

MÉTODO

restricciones: «no se puede modificar el código» y «menor esfuerzo operativo»

- A modifica el código → fuera
- C no resuelve el problema, solo lo retrasa; y en la clase 262 se vio que escalar puede empeorarlo
- D cumple, pero añade una máquina que mantener → incumple «menor esfuerzo operativo»

RESPUESTA B

→ aquí dos restricciones descartan dos opciones cada una

Pregunta 4 · Elige DOS.

ENUNCIADO

«Un equipo necesita que un trabajo por lotes nocturno se ejecute con el menor coste posible. El trabajo puede reiniciarse sin problema si se interrumpe y debe acabar antes de las 06:00. Elige DOS opciones.»

- A capacidad interrumpible
- B instancias reservadas a 3 años
- C puntos de control para reanudar si se interrumpe

- D instancias bajo demanda en la región más cara
- E ejecutar el trabajo en horario laboral

MÉTODO

«puede reiniciarse sin problema» + «menor coste»
 → A, claramente
 y A sin C es frágil: si se interrumpe, vuelve a empezar
 y puede no acabar antes de las 06:00

RESPUESTA A y C

- las preguntas de dos opciones suelen pedir una decisión y su salvaguarda
- B es caro para un trabajo nocturno y compromete 3 años
 clase 270

Pregunta 5 · La que parece técnica y decide la operación.

ENUNCIADO

«Una empresa quiere migrar una base de datos de 4 TB a la nube. El sistema no puede estar parado más de 15 minutos. El equipo no tiene experiencia con replicación. ¿Qué enfoque eliges?»

- A exportar, copiar por red e importar
- B servicio gestionado de migración con replicación continua y conmutación final
- C montar replicación manual entre el origen y el destino
- D copia física por dispositivo enviado por mensajería

MÉTODO

restricciones: «máximo 15 minutos de parada» y «el equipo no tiene experiencia con replicación»

- A 4 TB de exportación e importación: mucho más de 15 minutos → fuera
- D igual, y además el envío tarda días → fuera
- C cumple los 15 minutos e incumple «sin experiencia» → fuera

RESPUESTA B

- la frase «el equipo no tiene experiencia en» está ahí para descartar, siempre
- y es la restricción de personas, que también decide en la vida real
 clase 272

Los datos de preparación de cuatro personas.

persona A	solo vídeos y lectura	62	no	suspenso (68 %)
persona B	videos + prácticas con análisis de fallos	74	sí	aprobado (81 %)
persona C	construyó un proyecto pequeño + prácticas	58	sí	aprobado (89 %)
persona D	volcados de preguntas	40	no	aprobado (74 %)

Y lo que pasó seis meses después:

prueba interna de escenarios NUEVOS, no vistos

persona B	78 %
persona C	85 %
persona D	41 %

- quien memorizó volcados aprobó y no retuvo nada
- y además, esa práctica suele vulnerar las condiciones del examen

y en el trabajo, a los 12 meses
 persona C había construido tres cosas con lo estudiado y era la única que aplicaba el catálogo sin consultar

Y el análisis de fallos de la persona B, que fue lo que la hizo aprobar:

de 180 preguntas de práctica, falló 47	
clasificadas por causa	
no conocía el servicio	9
no leí una restricción	19
elegí lo que haría yo, no la doctrina	11
leí mal «elige DOS»	5
error de tiempo, respondí sin leer	3

- 38 de 47 fallos NO eran de conocimiento
- y los corrigió con método, no con más estudio
- segundo simulacro: 12 fallos de 180

La lección que esta clase deja: de 47 fallos en las preguntas de práctica, 38 no eran de conocimiento sino de método —restricciones no leídas, doctrina propia en vez de la del proveedor, enunciados de dos opciones mal leídos—, y se corrigieron sin estudiar más. Y quien memorizó volcados aprobó con un 74 % y a los seis meses resolvió el 41 % de escenarios nuevos.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-22-specializations-certifications-career/274-preguntas-de-escenario-y-estrategia-de-examen/lab.py
```

El laboratorio selecciona el motor de práctica assessment y produce labresult.json. El escenario, sus comprobaciones y el artefacto esperado corresponden a esta clase; no requiere credenciales y deja explícito qué debe revalidarse en un sandbox real.

1. Lee exercise.steps y formula una predicción antes de ejecutar.
2. Ejecuta la práctica y verifica todos los elementos de checks.
3. Provoca el caso de negativetest y explica la señal observada.
4. Materializa exam-simulation en evidence/ usando la plantilla indicada.
5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

El artefacto principal es una evaluación por escenarios con rúbrica y evidencia trazable. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye exam-simulation para el caso CloudShop. Incluye una alternativa descartada, un supuesto que pueda falsarse, una prueba de fallo y una decisión de rollback.

■ Criterio de aceptación

- [] lab.py termina con código 0 y genera JSON válido.
- [] La entrega conecta al menos tres requisitos con mecanismos verificables.
- [] Existe una prueba positiva y una prueba negativa con evidencia.
- [] Seguridad, costo y operación aparecen como decisiones, no como anexos.
- [] Se declara una limitación y una condición que obligaría a revisar el diseño.
- [] Otra persona puede repetir el recorrido sin conocimiento tácito.

■ Errores frecuentes

Síntoma	Causa probable	Corrección
Se falla eligiendo una opción que funcionaría en la vida real	El examen pregunta la doctrina del proveedor, no tu mejor solución	Descarta por incumplimiento de restricciones y, entre lo que quede, elige lo gestionado, nativo, con identidad y con menos piezas.
Se acaba el tiempo con preguntas sin responder	Se leyó el contexto entero antes que la pregunta y se agotó tiempo en dudas	Lee la pregunta final primero, haz tres pasadas, no pases de dos minutos por pregunta y responde siempre: no hay penalización.
Se aprueba y a los seis meses no se recuerda nada	Conocimiento inerte: se memorizó sin aplicar	Construye algo con lo estudiado y escríbelo; y elige el examen que se solapa con lo que haces ahora.
Se estudian muchos servicios y caen preguntas de límites y precios	Se estudia el catálogo y no las cuotas ni los modelos de coste	Dedica tiempo explícito a límites, cuotas y precios; es lo que más cae y lo que menos se estudia.
Se cambian respuestas al final y baja la nota	Se revisan también las preguntas contestadas con seguridad	Revisa solo lo que marcaste como dudoso; cambiar respuestas seguras empeora más veces de las que mejora.
Se tiene la certificación y no se pasa la entrevista técnica	El examen no puede medir juicio, diagnóstico ni decisión con información incompleta	Acompaña la certificación con evidencia de trabajo real: efecto, mecanismo y cifra.

■ Seguridad, ética y costo

Trabaja con cuentas propias o sandboxes autorizados. No publiques secretos, identificadores reales ni datos personales. Antes de crear recursos pagos define presupuesto, etiquetas y comando de destrucción; después verifica que no queden recursos huérfanos. Los ejemplos locales enseñan contratos, pero no certifican cumplimiento ni disponibilidad de producción.

■ Preguntas de comprobación

1. ¿Cuál es el orden correcto para leer una pregunta de escenario?
2. ¿Qué restricciones aparecen más y qué elimina cada una?
3. ¿Cómo se reconocen los distractores más frecuentes?
4. ¿Qué estrategia de pasadas y de tiempo funciona mejor?
5. ¿Qué mide un examen y qué hay que demostrar en otro sitio?

Referencias

- AWS (2024). Certification exam guides and sample questions. <<https://aws.amazon.com/certification/>>
- Microsoft (2024). Certification exam study guides and practice assessments. <<https://learn.microsoft.com/credentials/certifications/>>
- Google Cloud (2024). Certification exam guides. <<https://cloud.google.com/learn/certification>>
- Roediger, H. y Karpicke, J. (2006). Test-enhanced learning — por qué practicar preguntas supera a releer. <<https://journals.sagepub.com/doi/10.1111/j.1467-9280.2006.01693.x>>
- Haladyna, T. y Rodriguez, M. (2013). Developing and Validating Test Items — cómo se construyen los distractores. <<https://www.routledge.com/Developing-and-Validating-Test-Items/Haladyna-Rodriguez/p/book/9780415876049>>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 22 en PDF · Manual integral

Anterior	Índice	Siguiente
← 273 · Mapeo AWS, Azure, Google Cloud, Kubernetes y FinOps	Parte 22 · Programa	275 · Portafolio, evidencia, README y entrevista de sistemas →

Evaluación — 274 Preguntas de escenario y estrategia de examen

Preguntas

1. Explica el problema que resuelve esta capacidad sin mencionar un proveedor.
2. Identifica una frontera de responsabilidad y su propietario.
3. Compara dos alternativas mediante confiabilidad, seguridad, costo y operación.
4. ¿Qué demuestra el laboratorio y qué no puede demostrar?
5. Propón un caso límite o una prueba de fallo.

Reto verificable

Entrega exam-simulation con diagrama o configuración, evidencia de ejecución, alternativa descartada, riesgo residual y criterio de rollback.

Criterio de aceptación

- ☐ Resultado reproducible por una segunda persona.
- ☐ Evidencia vinculada a cada afirmación técnica.
- ☐ Prueba negativa o de fallo incluida.
- ☐ Costos y permisos expresados con unidades y alcance.
- ☐ Limitaciones declaradas sin presentar la simulación como producción.

Escala

Nivel	Evidencia
A — competente	Cumple todo y defiende trade-offs con datos.
B — en desarrollo	Cumple lo esencial; falta profundidad en una dimensión.
C — inicial	Artefacto parcial o conclusión sin evidencia suficiente.
0	Sin entrega reproducible.

275 — Portafolio, evidencia, README y entrevista de sistemas

Parte: 22 — Especializaciones, certificaciones y práctica profesional Nivel: intermedio-avanzado · Horas estimadas: 4
Laboratorio: assessment · Estado: EXECUTABLECORE

Propósito

Convertir el trabajo hecho en evidencia que otro puede evaluar en cinco minutos, y resolver una entrevista de diseño de sistemas. La clase da el formato de evidencia que funciona —efecto, mecanismo y cifra—, cómo se escribe un archivo de proyecto que se lee, y el método de la entrevista de diseño, donde lo que se evalúa no es la solución sino cómo se llega a ella.

Resultados de aprendizaje

Al finalizar podrás:

1. Escribir evidencia con efecto, mecanismo y cifra.
2. Estructurar un archivo de proyecto que alguien lee en cinco minutos.
3. Conducir una entrevista de diseño de sistemas con método.
4. Responder preguntas de comportamiento con hechos y no con adjetivos.
5. Elegir qué mostrar y qué omitir según a quién evalúa.

Conceptos centrales

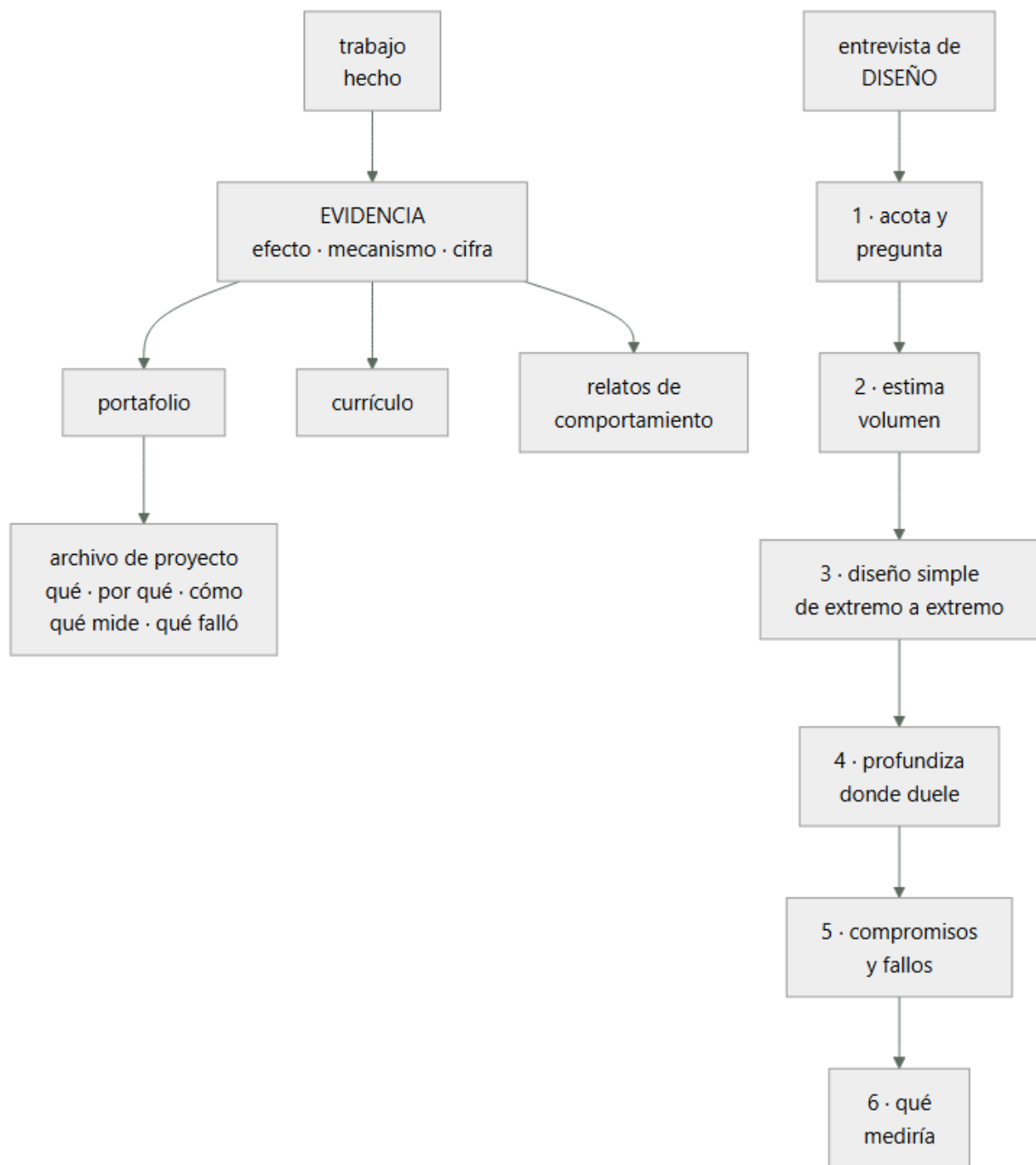
Concepto	Comprensión verificable
evidencia	Afirmación verificable sobre trabajo real: qué cambió, cómo y cuánto.
efecto, mecanismo y cifra	Formato mínimo de una afirmación creíble: qué mejoró, por qué mejoró y cuánto.
archivo de proyecto	Documento de entrada de un repositorio. Su primer párrafo decide si alguien sigue leyendo.
entrevista de diseño	Ejercicio abierto donde se evalúa el método, las preguntas y los compromisos, no la respuesta.
compromiso declarado	Decir qué se empeora al elegir. Su ausencia es la señal más clara de nivel bajo.
relato con hechos	Respuesta de comportamiento con situación, acción propia y resultado medido.

Modelo mental

Una especialización combina fundamentos, evidencia de proyectos y juicio bajo restricciones; una insignia sin práctica no sustituye esa combinación.

Aplicado a esta clase, separa siempre cuatro planos: intención (qué necesita el usuario), configuración (qué declaramos), estado observado (qué existe de verdad) y evidencia (cómo sabemos que cumple). Confundirlos produce diseños que se ven correctos en un diagrama pero fallan al operar.

■ Flujo de razonamiento



Desarrollo

1. Evidencia: efecto, mecanismo y cifra

La mayoría de los currículos y portafolios describen actividad. Lo que se evalúa es efecto.

ACTIVIDAD (no vale)

«responsable de la infraestructura en la nube»
 «implanté una plataforma interna»
 «trabajé con contenedores, canales de datos y monitorización»
 → no dice qué cambió ni si funcionó

EVIDENCIA (vale)

«bajé el tiempo de recuperación de 47 a 11 minutos reduciendo el lote de 14 cambios por despliegue a 1»
 «la adopción de la plataforma pasó de 3 a 38 de 41 servicios al bajar las pruebas de 40 a 9 minutos y la

configuración de 11 ficheros a 1»
«encontré que 19 de 34 procedimientos estaban rotos
ejecutándolos, y los convertí en ejecutables»

- EFECTO: qué mejoró, medido
- MECANISMO: por qué mejoró
- CIFRA: cuánto

Y por qué el mecanismo es la parte que más discrimina:

sin mecanismo, la cifra no se puede evaluar
«mejoré el rendimiento un 40 %» → ¿cómo? ¿desde qué
base? ¿a costa de qué?

con mecanismo, la conversación se vuelve técnica
→ y ahí es donde alguien de nivel 3 se distingue

y el mecanismo delata a quien no hizo el trabajo
→ quien lo hizo puede explicarlo tres niveles hacia
abajo

Y qué hacer si no hay cifras:

CASI SIEMPRE HAY ALGO CONTABLE
cuántos, cuánto tardaba, cuántas veces al mes, de qué a
qué
→ «de no poder restaurar a restaurar en 52 minutos,
probado mensualmente»

y si de verdad no hay
se cuenta el CAMBIO CUALITATIVO con precisión
→ «antes nadie sabía qué recursos existían; ahora hay
inventario con dueño, generado cada 6 horas»

→ lo que no vale es el adjetivo: «mejoré mucho la
fiabilidad»

Y la evidencia del fracaso, que casi nadie incluye y que más impresiona:

«la primera remediación automática convirtió una
degradación en una caída en 3 minutos; le faltaba la
condición de mínimo sano. Desde entonces todo pasa por
modo sombra cuatro semanas»

- demuestra nivel 3 mejor que un éxito
- porque demuestra que hubo consecuencias y aprendizaje
- y porque nadie inventa un fracaso con ese detalle

2. El archivo de proyecto

Un repositorio se juzga por su archivo de entrada, y casi siempre en menos de un minuto.

LA ESTRUCTURA QUE FUNCIONA

- 1 QUÉ ES, en dos frases
y para quién
- 2 EL PROBLEMA que resuelve, concreto
- 3 CÓMO SE PRUEBA
→ los comandos para verlo funcionando
→ y esto va ARRIBA, no al final
- 4 DECISIONES, con sus alternativas
→ «elegí X frente a Y porque Z; lo que empeora es W»
clase 272
- 5 QUÉ MIDE que funciona
- 6 QUÉ NO HACE y qué falta
- 7 y QUÉ APRENDÍ, incluido lo que salió mal

- el punto 4 es el que distingue un proyecto de
aprendizaje de un ejercicio copiado
- y el 6 genera confianza: quien dice lo que falta suele
saber de qué habla

Y los errores que descartan un portafolio:

el proyecto no arranca siguiendo las instrucciones
→ y esto se comprueba, y falla mucho más de lo que se
cree

credenciales en el repositorio
 → descarte inmediato en cualquier evaluación seria
 clase 197

tres proyectos idénticos de un tutorial
 → mejor uno propio, aunque sea pequeño

un README que explica la tecnología en vez del problema
 y cero pruebas o cero señales
 → en un programa como este, es incoherente

Y qué proyecto elegir, si hay que construir uno:

MEJOR PEQUEÑO Y COMPLETO QUE GRANDE E INACABADO

un solo servicio con

infraestructura como código	clase 128
cadena de entrega con despliegue progresivo	clase 102
identidad sin credenciales permanentes	clase 256
señales e indicadores de servicio	clase 268
copias con restauración probada	clase 255
y coste medido	clase 270

→ eso demuestra más que un sistema distribuido a medias
 → y permite responder «¿qué pasa si...?» con datos

Y la adaptación a quien evalúa:

filtro automático o de recursos humanos
 → palabras del anuncio, cifras claras, sin jerga

ingeniería
 → mecanismos y compromisos

dirección
 → efecto en negocio: coste, riesgo, tiempo de entrega

→ el mismo trabajo, tres presentaciones
 → y mentir en cualquiera se detecta en la siguiente conversación

3. La entrevista de diseño de sistemas

El error más común es empezar a dibujar. Lo que se evalúa es el método.

QUÉ SE EVALÚA DE VERDAD

qué preguntas haces antes de decidir

si acotas el problema o lo aceptas tal cual

si estimas volúmenes

si declaras compromisos

si anticipas fallos

y si dices qué medirías

→ no hay respuesta correcta
 → y quien empieza a dibujar sin preguntar ya ha perdido la mitad

Y el método, con tiempos aproximados para 45 minutos:

- 1 ACOTAR Y PREGUNTAR 5-8 min
 - ¿quiénes son los usuarios y qué hacen?
 - ¿cuántos y con qué crecimiento?
 - ¿qué es lo crítico: latencia, coherencia, coste?
 - ¿qué está fuera de alcance?
 - y escribe los requisitos donde se vean
- 2 ESTIMAR 3-5 min
 - peticiones por segundo, tamaño de los datos, crecimiento
 - con números redondos y dichos en voz alta
 - «un millón de usuarios activos al día, 20 acciones cada uno, son 20 millones al día, unas 230 por segundo de media y quizá 700 en pico»
- 3 DISEÑO SIMPLE DE EXTREMO A EXTREMO 8-10 min
 - la versión que funciona, sin adornos
 - y decir explícitamente «esto es la versión simple; ahora veamos dónde se rompe»
- 4 PROFUNDIZAR DONDE DUELE 10-15 min

el modelo de datos y su partición clase 208
el camino crítico y sus plazos clase 201
qué pasa cuando una dependencia va lenta clase 185
y la coherencia que se necesita de verdad

5 COMPROMISOS Y FALLOS 5-8 min
qué empeora cada elección
qué se cae si cae una zona
y qué se degrada antes que caer clase 262

6 QUÉ MEDIRÍA 3-5 min
indicadores por recorrido, no por máquina clase 268
→ y con esto se cierra muy bien

Y las señales que buscan quienes entrevistan:

SEÑALES BUENAS

pregunta antes de decidir
dice números en voz alta
empieza simple y complica con motivo
 nombra lo que empeora sin que se lo pidan
dice «no lo sé» y propone cómo averiguarlo
y cambia de opinión cuando le dan un dato nuevo

SEÑALES MALAS

dibuja una arquitectura elaborada de entrada
añade componentes sin justificarlos
no estima nada
defiende su primera idea contra la evidencia
y no menciona fallos ni operación

→ estas dos listas son las mismas de la clase 272,
aplicadas a 45 minutos

4. Preguntas de comportamiento y qué omitir

La otra mitad de la entrevista, que decide más de lo que se admite.

EL FORMATO QUE FUNCIONA

SITUACIÓN contexto en dos frases, con números
TAREA qué había que conseguir
ACCIÓN qué hiciste TÚ, no el equipo
RESULTADO medido, y qué aprendiste

→ y la parte que más se falla es la acción: se cuenta en plural y no se sabe qué hizo la persona

Y las preguntas que más se repiten, con lo que buscan:

«cuéntame un incidente que dirigiste»
→ buscan método, comunicación y qué cambiaste después

«una decisión técnica de la que te arrepientes»
→ buscan honestidad y capacidad de análisis
→ responder «ninguna» es la peor respuesta posible

«un desacuerdo técnico con alguien»
→ buscan cómo se resuelve, no quién tenía razón
→ y la mejor respuesta suele incluir «tenía razón la otra persona y así lo comprobamos»

«algo que hiciste que nadie te pidió»
→ buscan iniciativa con criterio

y «cómo te mantienes al día»
→ buscan método, no lista de fuentes

Y qué omitir, que también es una decisión:

NO PONGAS

tecnologías que no sabrías defender
→ se pregunta por lo que está escrito
proyectos de tutorial sin aportación propia
niveles autoevaluados en barras de porcentaje
→ nadie sabe qué significa un 80 % en redes

y trabajo de otros presentado como propio

SÍ PON

lo que puedas explicar tres niveles hacia abajo
y una cosa que salió mal, con lo que cambiaste

Y la lista de comprobación de la clase:

- cada afirmación tiene efecto, mecanismo y cifra
- puedo explicar el mecanismo tres niveles hacia abajo
- incluyo al menos un fracaso con su corrección
- mi archivo de proyecto dice el problema antes que la tecnología
- los comandos para verlo funcionando están arriba
- hay una sección de decisiones con alternativas
- hay una sección de qué no hace y qué falta
- el proyecto arranca siguiendo las instrucciones
- no hay credenciales en ningún repositorio
- en diseño, pregunto y estimo antes de dibujar
- empiezo simple y digo dónde se rompe
- nombro lo que empeora sin que me lo pidan
- cierro diciendo qué mediría
- mis relatos dicen qué hice yo, con resultado medido

Y el cierre que enlaza con la clase siguiente: con la evidencia escrita y el método de diseño, queda ponerlo a prueba delante de gente que pregunta. La defensa técnica ante un panel es el proyecto de la clase 276, que cierra la parte 22.

Ejemplo trabajado

Tres transformaciones de evidencia y una entrevista de diseño completa. Lo que sigue es el mismo trabajo escrito de tres formas, el archivo de proyecto que pasó de descartado a entrevista, y la transcripción resumida de una entrevista de diseño con lo que el evaluador anotó.

Transformación 1 · El mismo trabajo, tres versiones.

VERSIÓN 1 · actividad

«Responsable de la operación de la plataforma en la nube. Implanté monitorización, procedimientos y mejoras en el proceso de despliegue.»

VERSIÓN 2 · con cifras, sin mecanismo

«Reducí los incidentes un 62 % y el tiempo de recuperación un 77 %.»

→ mejor, y no se puede evaluar

→ «¿cómo?» y no hay respuesta escrita

VERSIÓN 3 · efecto, mecanismo y cifra

«Los incidentes graves pasaron de 34 a 13 al año y el tiempo de recuperación de 47 a 11 minutos. El mecanismo principal fue reducir el lote: de 14 cambios por despliegue a 1, sustituyendo el comité de aprobación —que rechazaba el 0,6 %— por cinco comprobaciones automáticas. Con lotes de un cambio, revertir es inmediato y no hay que averiguar cuál de catorce fue.»

Y lo que ocurrió en la entrevista con cada versión:

versión 1 3 preguntas genéricas, conversación corta

versión 3 la entrevista se fue 20 minutos por
 «cuéntame esas cinco comprobaciones»
 → y ahí se demostró el nivel

→ una evidencia bien escrita DIRIGE la entrevista hacia donde uno es fuerte

Transformación 2 · El archivo de proyecto.

ANTES

cloudshop-infra

Proyecto de infraestructura con Terraform, Kubernetes, Prometheus y GitHub Actions.

Instalación

(47 líneas de pasos manuales)

- descartado en 20 segundos
- dice tecnologías, no problema; y arrancar cuesta 47 pasos

Y la versión reescrita:

DESPUÉS

```
# Servicio de pedidos con operación completa

Un servicio de pedidos pequeño, montado con todo lo que
hace falta para operarlo de verdad: despliegue
progresivo, señales por recorrido, copias con
restauración probada y coste medido. Sirve para
enseñar cómo se opera un servicio, no cómo se escribe
uno.

## Verlo funcionando en 3 minutos
```
make up
make smoke
make chaos-kill-instance
```

El tercer comando mata una instancia; el servicio sigue
respondiendo y el panel lo muestra.

## El problema
La mayoría de los proyectos de ejemplo despliegan y
ahí acaban. Este responde a: ¿qué pasa cuando algo
falla, quién se entera y cuánto se tarda en volver?

## Decisiones
| decisión | alternativa | por qué | qué empeora |
|---|---|---|---|
| base relacional única | una por servicio | simplicidad operativa con 1 persona | escalado de
escritura |
| despliegue por porcentajes | azul-verde | vuelta atrás fina | despliegue más lento |
| señales por recorrido | por máquina | refleja al usuario | más trabajo de instrumentación |

## Qué mide que funciona
- éxito de compra > 99,5 %, medido en el cliente
- restauración probada: 4 min 12 s la última vez
- coste: 0,004 USD por pedido

## Qué NO hace
- no hay multi-región: el objetivo de recuperación
asumido son 4 horas
- no hay autenticación de usuario final
- la caché es en memoria y se pierde al reiniciar

## Qué aprendí
La primera versión del despliegue progresivo retiraba
instancias sin comprobar cuántas sanas quedaban. Con 3
de 4 fallando por una dependencia lenta, las retiró
todas. Añadí la condición de mínimo sano y una
detección de fallo general.
```

Y el resultado:

candidaturas enviadas con la versión antigua	14
entrevistas	1
con la versión nueva	11
entrevistas	6
y en 5 de las 6, el entrevistador citó la sección	
«Qué aprendí»	

→ el fracaso contado con detalle fue lo más citado

Una entrevista de diseño, resumida.

ENUNCIADO

«Diseña el sistema de seguimiento de pedidos de una tienda: el usuario quiere ver dónde está su pedido.»

MINUTO 0-6 · ACOTAR

«¿cuántos pedidos activos hay a la vez?» ~2 M
«¿de dónde viene el estado?» de 4 transportistas,
por API y por fichero
«¿con qué frescura debe verse?» «unos minutos está
bien»
«¿el usuario consulta o quiere que le avisen?»
ambas
«¿qué queda fuera?» devoluciones

y el candidato escribió:
2 M pedidos activos · 4 orígenes · frescura de
minutos · consulta y aviso

MINUTO 6-10 · ESTIMAR, en voz alta

«2 M pedidos activos, con 3 actualizaciones al día cada
uno, son 6 M eventos/día, unos 70 por segundo de media
y quizá 300 en pico»
«si el usuario consulta 2 veces por pedido, son 4 M
consultas/día, unas 46 por segundo»
«el estado de un pedido son unos 200 bytes; 2 M
activos son 400 MB: cabe en memoria si hiciera falta»

→ el evaluador anotó: «estima sin que se lo pidan y usa
la estimación para decidir»

MINUTO 10-19 · DISEÑO SIMPLE

ingesta de los 4 orígenes → cola → procesador →
almacén de estado → API de consulta
y avisos por eventos desde el procesador

«esta es la versión simple; ahora dónde se rompe»

MINUTO 19-33 · DONDE DUELE

el evaluador pregunta: «un transportista envía eventos
desordenados y a veces repetidos»
→ «entonces necesito idempotencia y orden por pedido:
clave de partición el identificador de pedido, y
descarto eventos con marca anterior a la última
aplicada» clase 210

«uno de los transportistas deja de responder»
→ «el estado se queda viejo. Prefiero mostrar la
antigüedad –(actualizado hace 3 horas)– a mostrar
un estado falso. Y una alerta de frescura por
transportista, no global, porque global se diluye»
clase 268

«los ficheros de un transportista llegan con 6 horas de
retraso una vez»

→ «marca de agua y reproceso; el estado se corrige y
el aviso ya enviado no se puede deshacer, así que
los avisos solo se disparan con eventos de las APIs
en tiempo real» clase 242

MINUTO 33-40 · COMPROMISOS

«lo que empeora con la partición por pedido: un pedido
con muchísimos eventos crea una partición caliente.
Con 3 al día no pasa, pero si añadieran geolocalización
continua, sí; ahí lo revisaría» clase 208
«y no tengo coherencia fuerte entre el estado y el
aviso: puede llegar el aviso antes de que la consulta
lo muestre. Lo acepto y lo mitigo leyendo de la réplica
primaria durante 30 segundos tras un aviso»

MINUTO 40-45 · QUÉ MEDIRÍA

«frescura por transportista, percentil 95»
«éxito de la consulta de estado»
«retraso de la cola»
«y avisos enviados con estado incorrecto, que es el
fallo que más molesta al usuario y el que menos se
mide»

Y lo que el evaluador anotó al final:

- + acotó antes de dibujar y escribió los requisitos
- + estimó en voz alta y usó la estimación
- + empezó simple y complicó con motivo
- + mencionó lo que empeora sin que se lo pidieran
- + propuso mostrar la antigüedad en vez de un estado falso
→ «esa respuesta la dan pocos y dice mucho»
- + cerró con una métrica que nadie suele nombrar
- no habló de coste en ningún momento
- no preguntó por normativa ni por datos personales

contratado

Y la observación del propio candidato después:

«no dije nada que no estuviera en el programa; lo único que hice distinto fue no dibujar hasta el minuto diez»

La lección que esta clase deja: la misma experiencia escrita como actividad dio 1 entrevista en 14 candidaturas y escrita con efecto, mecanismo y cifra dio 6 en 11 —y en cinco de esas seis, lo que el entrevistador citó fue la sección donde se contaba un fracaso con detalle. Y en la entrevista de diseño, lo que el evaluador subrayó no fue ninguna pieza del sistema: fue que no se dibujó nada hasta el minuto diez.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-22-specializations-certifications-career/275-portafolio-evidencia-readme-y-entrevista-de-sistemas/lab.py
```

El laboratorio selecciona el motor de práctica assessment y produce labresult.json. El escenario, sus comprobaciones y el artefacto esperado corresponden a esta clase; no requiere credenciales y deja explícito qué debe revalidarse en un sandbox real.

1. Lee exercise.steps y formula una predicción antes de ejecutar.
2. Ejecuta la práctica y verifica todos los elementos de checks.
3. Provoca el caso de negativetest y explica la señal observada.
4. Materializa professional-portfolio en evidence/ usando la plantilla indicada.
5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

El artefacto principal es una evaluación por escenarios con rúbrica y evidencia trazable. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye professional-portfolio para el caso CloudShop. Incluye una alternativa descartada, un supuesto que pueda falsarse, una prueba de fallo y una decisión de rollback.

■ Criterio de aceptación

- [] lab.py termina con código 0 y genera JSON válido.
- [] La entrega conecta al menos tres requisitos con mecanismos verificables.
- [] Existe una prueba positiva y una prueba negativa con evidencia.
- [] Seguridad, costo y operación aparecen como decisiones, no como anexos.
- [] Se declara una limitación y una condición que obligaría a revisar el diseño.
- [] Otra persona puede repetir el recorrido sin conocimiento tácito.

■ Errores frecuentes

Síntoma	Causa probable	Corrección
El currículum describe responsabilidades y no genera entrevistas técnicas	Describe actividad en vez de efecto	Reescribe cada punto con efecto, mecanismo y cifra; el mecanismo es lo que convierte la conversación en técnica.
El portafolio se descarta en segundos	El archivo de entrada habla de tecnologías y arrancar cuesta decenas de pasos	Pon el problema en las dos primeras frases y los comandos para verlo funcionando arriba; comprueba que arranca siguiendo tus instrucciones.
En la entrevista de diseño la conversación se pierde	Se empezó a dibujar sin acotar ni estimar	Dedica los primeros minutos a preguntar y escribir requisitos, estima en voz alta y empieza por la versión simple diciendo dónde se rompe.
Preguntan por algo del currículum y no se sabe explicar	Se listó tecnología que no se puede defender	Pon solo lo que puedas explicar tres niveles hacia abajo; lo demás resta credibilidad al resto.
Las respuestas de comportamiento suenan vacías	Se cuentan en plural y sin resultado medido	Di qué hiciste tú, con contexto numérico y resultado; y ten preparada una decisión de la que te arrepientes, con su análisis.
El proyecto es grande, impresionante y no se puede probar	Se priorizó tamaño sobre completitud	Mejor un servicio pequeño con despliegue, señales, copias probadas y coste medido: permite responder a qué pasa si algo falla.

■ Seguridad, ética y costo

Trabaja con cuentas propias o sandboxes autorizados. No publiques secretos, identificadores reales ni datos personales. Antes de crear recursos pagos define presupuesto, etiquetas y comando de destrucción; después verifica que no queden recursos huérfanos. Los ejemplos locales enseñan contratos, pero no certifican cumplimiento ni disponibilidad de producción.

■ Preguntas de comprobación

1. ¿Qué tres partes tiene una afirmación de evidencia creíble y cuál discrimina más?
2. ¿Qué secciones hacen que un archivo de proyecto se lea y genere confianza?
3. ¿Qué se evalúa realmente en una entrevista de diseño de sistemas?
4. ¿Por qué contar un fracaso con detalle funciona mejor que otro éxito?
5. ¿Qué conviene omitir de un currículum o portafolio y por qué?

Referencias

- Xu, A. (2020). System Design Interview: an insider's guide. <<https://www.oreilly.com/library/view/system-design-interview/9798641135823/>>
- Google (2024). How we hire: interviewing at Google. <<https://careers.google.com/how-we-hire/>>
- Amazon (2024). Interview process and behavioral questions guidance. <<https://amazon.jobs/content/en/how-we-hire/interviewing-at-amazon>>
- GitHub (2024). About READMEs. <<https://docs.github.com/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-readmes>>
- Kleppmann, M. (2017). Designing Data-Intensive Applications — base habitual de las entrevistas de diseño. <<https://dataintensive.net/>>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Anterior	Índice	Siguiente
← 274 · Preguntas de escenario y estrategia de examen	Parte 22 · Programa	276 · Proyecto: defensa técnica ante panel →

Evaluación — 275 Portafolio, evidencia, README y entrevista de sistemas

Preguntas

1. Explica el problema que resuelve esta capacidad sin mencionar un proveedor.
2. Identifica una frontera de responsabilidad y su propietario.
3. Compara dos alternativas mediante confiabilidad, seguridad, costo y operación.
4. ¿Qué demuestra el laboratorio y qué no puede demostrar?
5. Propón un caso límite o una prueba de fallo.

Reto verificable

Entrega professional-portfolio con diagrama o configuración, evidencia de ejecución, alternativa descartada, riesgo residual y criterio de rollback.

Criterio de aceptación

- ☐ Resultado reproducible por una segunda persona.
- ☐ Evidencia vinculada a cada afirmación técnica.
- ☐ Prueba negativa o de fallo incluida.
- ☐ Costos y permisos expresados con unidades y alcance.
- ☐ Limitaciones declaradas sin presentar la simulación como producción.

Escala

Nivel	Evidencia
A — competente	Cumple todo y defiende trade-offs con datos.
B — en desarrollo	Cumple lo esencial; falta profundidad en una dimensión.
C — inicial	Artefacto parcial o conclusión sin evidencia suficiente.
0	Sin entrega reproducible.

276 — Proyecto: defensa técnica ante panel

Parte: 22 — Especializaciones, certificaciones y práctica profesional Nivel: intermedio-avanzado · Horas estimadas: 8
Laboratorio: capstone · Estado: EXECUTABLECORE

Propósito

Defender una decisión técnica ante un panel que pregunta, con el material de toda la parte 22. La clase da el encargo, el formato, la rúbrica con la que se evalúa y las pruebas negativas. Y cierra la parte 22: corrige las cinco predicciones de la clase 264 —tres acertadas, una acertada por el motivo equivocado y una fallida—, actualiza el recuento de leyes, añade la ley 31 y escribe la hipótesis de la parte 23.

Resultados de aprendizaje

Al finalizar podrás:

1. Preparar una defensa técnica con decisión, alternativas y criterio.
2. Responder preguntas hostiles sin defender la decisión, sino el criterio.
3. Evaluar una defensa con una rúbrica explícita.
4. Corregir las cinco predicciones de la clase 264 con evidencia.
5. Escribir la hipótesis de la parte 23 en forma refutable.

Conceptos centrales

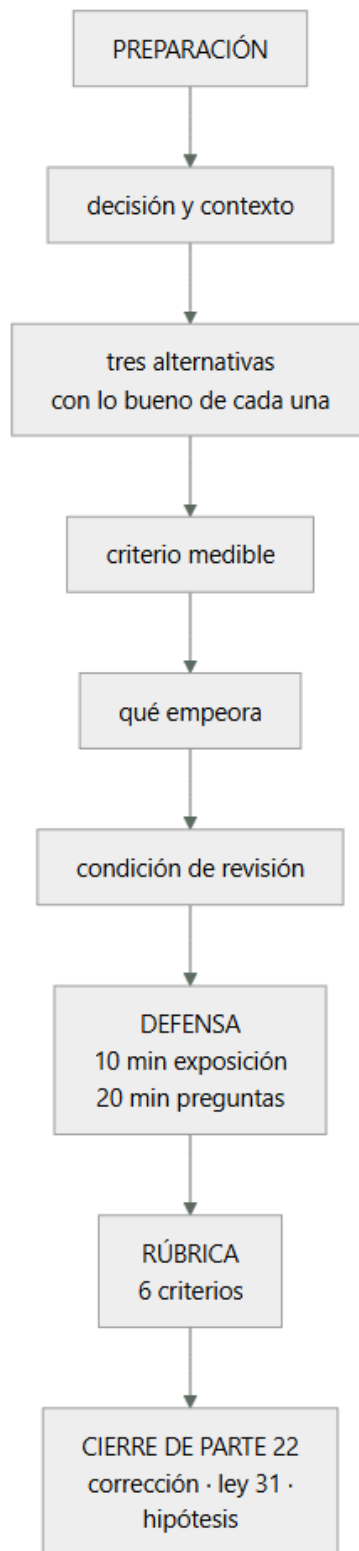
Concepto	Comprensión verificable
defensa técnica	Exposición breve de una decisión seguida de preguntas abiertas de un panel. Se evalúa el razonamiento, no la solución.
ley 31	Toda función de apoyo tiende a convertirse en la que autoriza; y al hacerlo, deja de ver el sistema real.
rúbrica	Criterios explícitos de evaluación, conocidos antes. Convierte una impresión en un juicio.
pregunta hostil	La que busca el punto débil real. Su respuesta correcta suele empezar por reconocerlo.
alcance frente a visibilidad	Distinción que decide una carrera: contar el trabajo no es lo mismo que hacer trabajo que afecta a otros.
hipótesis de parte	Afirmación refutable escrita antes de estudiar, que la parte siguiente corrige con evidencia.

Modelo mental

Una especialización combina fundamentos, evidencia de proyectos y juicio bajo restricciones; una insignia sin práctica no sustituye esa combinación.

Aplicado a esta clase, separa siempre cuatro planos: intención (qué necesita el usuario), configuración (qué declaramos), estado observado (qué existe de verdad) y evidencia (cómo sabemos que cumple). Confundirlos produce diseños que se ven correctos en un diagrama pero fallan al operar.

■ Flujo de razonamiento



Desarrollo

1. El encargo, el formato y la rúbrica

El encargo. Elige una decisión técnica real —tuya o del programa— y defiéndela ante un panel de tres personas: una técnica de tu área, una técnica de otra área y una que representa a negocio.

EL FORMATO

10 minutos de exposición

20 minutos de preguntas
5 minutos de cierre: qué cambiarías tras la conversación

→ y ese último tramo vale tanto como los otros dos

Lo que hay que llevar preparado, que es el material de la clase 272:

- 1 la decisión, en una frase
- 2 el contexto y las restricciones reales
- 3 los atributos de calidad, ORDENADOS
- 4 tres alternativas, con lo bueno de cada una dicho en serio
- 5 el criterio medible que decidió
- 6 qué empeora con la elección
- 7 la condición de revisión
- 8 y el coste de la alternativa descartada clase 270

La rúbrica, conocida antes de empezar:

1	ACOTACIÓN	
	¿queda claro el problema y lo que queda fuera?	15
2	ALTERNATIVAS	
	¿son tres reales? ¿se presenta bien lo bueno de las descartadas?	20
3	CRITERIO	
	¿es medible y se aplicó de verdad?	20
4	COMPROMISOS	
	¿dice qué empeora sin que se lo pidan?	20
5	RESPUESTA A LAS PREGUNTAS	
	¿defiende el criterio y no la decisión?	
	¿dice «no lo sé» cuando toca?	15
6	REVISIÓN	
	¿hay condición que la invalide?	10

→ y suspenden casi siempre por el 2 y el 4

→ el 2 porque llevan una alternativa de paja

→ el 4 porque presentan la decisión como sin coste

Y las preguntas que el panel debe hacer:

«¿qué te haría cambiar de opinión?»
«¿cuál es la opción más barata y por qué no la elegiste?»
«¿qué pasa si el volumen se multiplica por diez?»
«¿quién opera esto y con qué guardia?» clase 268
«¿qué se rompe primero?» clase 262
«¿cuánto cuesta al mes y de dónde sale esa cifra?»
«si te equivocaste, ¿cuándo lo sabrás?»
y «¿qué parte de esto no has probado?»

→ la última desarma a quien ha memorizado una defensa

2. Las pruebas negativas de la parte

Cada clase dejó una comprobación. Aquí se ejecutan sobre uno mismo.

- clase 265 coge tres competencias que dices tener
→ ¿qué falló alguna vez en cada una?
→ si en alguna es «nada», tu nivel ahí es 1
- clase 266 ¿cuál es tu tiempo de entrega y qué tramo lo domina?
→ y ¿cuándo fue tu última vuelta atrás?
- clase 267 ¿por qué el último equipo que no adoptó tu herramienta no la adoptó?
→ si no lo sabes, no es un producto
- clase 268 coge tus objetivos de servicio
→ ¿alguno se incumple alguna vez?
→ si todos se cumplen siempre, no miden nada
- clase 269 haz algo que debería detectarse y cronometra
→ ¿cuánto tardó? ¿alguien llamó?
- clase 270 ¿cuál es tu coste por unidad de negocio y su

tendencia?
→ y ¿cuánto coste evitado has registrado?

clase 271 ¿de dónde salieron los casos de tu evaluación?
→ y ¿cómo detectas un dato malo que no da error?

clase 272 coge tu última decisión importante
→ ¿está escrita con alternativas y condición de revisión?

clase 273 coge un servicio que uses
→ ¿sabrías traducirlo por restricción a otra nube y decir dónde se rompe la equivalencia?

clase 274 haz 20 preguntas de práctica y clasifica tus fallos
→ ¿cuántos son de conocimiento y cuántos de método?

clase 275 dale tu archivo de proyecto a alguien
→ ¿arranca siguiendo tus instrucciones?
→ ¿entiende el problema en dos frases?

Y la prueba que resume la parte entera:

LA PRUEBA DE LOS CINCO MINUTOS

cuenta a alguien que no conoce tu trabajo
qué cambió por lo que hiciste
por qué mecanismo
con qué cifra
y qué salió mal por el camino

→ en cinco minutos y sin jerga
→ si no puedes, no es que no lo hayas hecho: es que no sabes qué hiciste

3. Corrección de las cinco predicciones de la clase 264

La corrección, con evidencia y sin adornos.

1. «las certificaciones correlacionarán con conseguir entrevistas y no con resolver problemas; y el hueco estará donde no se puede examinar por opción múltiple: decidir con información incompleta»

CORRECTA, E INJUSTA EN UN PUNTO. Los datos la sostienen: quien memorizó volcados aprobó con un 74 % y seis meses después resolvió el 41 % de escenarios nuevos. Pero fuimos injustos con lo que el examen SÍ mide: límites, cuotas y modelos de precio, que es lo que más cae y lo que menos se estudia, y que en la clase 262 resultó ser la causa de que un plan de continuidad cubriera el 3,2 % de la capacidad prometida. Eso no es catálogo inútil.

2. «la especialización que peor envejecerá será la atada a un producto, y la que mejor, la atada a una RESTRICCIÓN, porque las restricciones sobreviven a los productos»

CORRECTA Y DEMOSTRADA CON UN MÉTODO. La clase 273 lo convirtió en procedimiento: traducir por restricción encontró cuatro diferencias que la tabla por nombres no recogía, dos de ellas silenciosas. Y la evidencia inversa: la política traducida literalmente pasó la revisión humana concediendo acceso a 214 conjuntos de datos en vez de uno.

3. «buena parte del consejo de carrera tratará de visibilidad y no de destreza; será incómodo y será cierto»

FALLIDA EN EL DIAGNÓSTICO. Predijimos visibilidad y lo que apareció fue otra cosa: la diferencia entre quien

avanza y quien no no era contar el trabajo, era que el trabajo tuviera EFECTO SOBRE OTROS. La persona con ocho años y nivel 3 estable no tenía un problema de visibilidad: llevaba dos años sin cambiar cómo trabajaba ningún otro equipo. Y lo que convirtió una candidatura de una entrevista en catorce a seis en once no fue promocionarse: fue poder nombrar el MECANISMO, que requiere haberlo producido. Confundimos alcance con visibilidad.

4. «lo que el mercado paga en multinube no será saber tres nubes: será saber una a fondo y poder TRADUCIR»

CORRECTA, Y CON UN MECANISMO QUE NO TENÍAMOS. Acertamos el qué y no sabíamos el porqué. El porqué es doble: la profundidad es lo que enseña la restricción, y la restricción es lo único que se traduce; y las empresas no quieren generalistas de tres nubes porque no quieren tres nubes. Operar la segunda le costaba a CloudShop 2,5 personas y 492.000 USD al año, y se mantuvo por un contrato concreto, no por estrategia.

5. «más de la mitad de lo enseñado será transferible entre nubes sin cambios, y menos de una quinta parte será sintaxis de un producto concreto; y los temarios de certificación invertirán esa proporción»

CORRECTA EN LA CIFRA Y CIEGA EN UN TERCIO. La clasificación del contenido dio 71 % de restricción, 18 % de producto y 11 % de organización y personas. Las dos cifras predichas se cumplen. Lo que no vimos es ese 11 %: no lo predijimos y resultó ser donde vive el modo de fracaso de las seis rutas. Ninguna especialidad fracasa por falta de técnica.

Marcador: tres correctas, una correcta por el motivo equivocado y una fallida. Y el fallo de la tercera enseña lo que obliga a escribir la ley: el 11 % que no predijimos —lo organizativo— es donde todas las rutas se estropean, y siempre de la misma forma.

4. Recuento de leyes, ley 31 e hipótesis de la parte 23

El recuento de leyes, cerrada la parte 22.

ley 13	lo que no se mira deja de funcionar en silencio	68
ley 15	la señal existe y nadie la mira	58
ley 22	un procedimiento nunca ejecutado no funciona	55
ley 16	un control que estorba se rodea	43
ley 14	el coste se decide al crear, no al pagar	41
ley 20	lo que no tiene dueño se filtra y se desperdicia	38
ley 25	lo provisional sobrevive a su motivo	34
ley 21	el acoplamiento vive en quién escribe	34
ley 26	el valor por defecto sirve a la demostración	29
ley 23	la capacidad la limita lo que ya se mantiene	25
ley 17	se optimiza la medida, no el objetivo	22
ley 24	lo que no está en el diagrama no se analiza	18
ley 27	un control solo actúa sobre lo que cambia	13
ley 19	la compensación hace invisible el fallo	11
ley 18	lo asíncrono traslada la garantía, no la elimina	11
ley 28	donde se paga por uso, cada defecto es una factura	11
ley 29	un fallo de datos no da error: da otro número	10
ley 30	automatizar elimina un juicio que nadie escribió	9
ley 31	la función de apoyo acaba autorizando, y se queda ciega	6

Y la ley que la parte 22 obliga a escribir:

LEY 31
toda función de apoyo tiende a convertirse en la que autoriza; y en cuanto lo hace, deja de ver el sistema real

apariciones en esta parte 6
clase 266 el equipo de entrega acabó aprobando todos los

- despliegues: rechazaba el 0,6 % y añadía 4,1 días de espera
- clase 267 una plataforma obligatoria apaga la señal de adopción y esconde sus defectos en vez de corregirlos
- clase 268 la variante inversa: absorber la guardia ajena hace que quien construye deje de sufrir lo que construye
- clase 269 bloquear sin alternativa produjo 1.107 recursos fuera del inventario, incluida una base con datos de clientes
- clase 270 medirse por el ahorro conseguido convierte la función en la que aprueba gastos, y le quita de la vista el coste evitado
- clase 272 la arquitectura de control revisa diseños ajenos y reproduce el comité de cambios

y lo que la distingue de la ley 16
 la 16 describe a QUIEN SUFRE el control: lo rodea
 la 31 describe al que LO EJERCE: deriva hacia autorizar porque parece prudente, y su castigo no es la lentitud
 → es la CEGUERA: deja de ver lo que realmente ocurre
 → y por eso la métrica que salva a una función de apoyo es siempre la misma: cuántas decisiones buenas se toman sin ella

La hipótesis de la parte 23 (clases 277 a 288, capstones sectoriales), escrita antes de estudiarla:

1. los ocho sectores diferirán mucho menos en tecnología de lo que aparentan: lo que cambiará será QUÉ RESTRICCIÓN MANDA y qué se acepta perder cuando algo falla
2. en todos ellos, la restricción dominante será NO TÉCNICA: normativa, contrato o coste unitario; y la técnica se acomodará a ella ley 23
3. el capstone que más fallos silenciosos revele será el de datos e inteligencia artificial, porque un fallo de datos no da error ley 29
4. en el ensayo integrado, alrededor de dos tercios de los hallazgos volverán a ser organizativos, como en la clase 261 leyes 30, 31
5. y una refutable con cifra: **más del 70 % de las decisiones de los ocho capstones se podrán justificar con leyes ya escritas, sin necesidad de una nueva**; y las que no, se concentrarán en sector público y salud, donde la restricción es jurídica y no física

Y el cierre de la parte 22: de doce clases, lo que más cambió resultados no fue técnico: fue que la misma experiencia, escrita como actividad, dio una entrevista en catorce candidaturas y escrita con efecto, mecanismo y cifra dio seis en once. La parte 23 baja de nuevo al sistema y lo somete a ocho sectores con restricciones distintas. Empieza por el comercio en varias regiones. Es la clase 277.

Ejemplo trabajado

Dos defensas ante panel, evaluadas con la rúbrica. Lo que sigue es la que suspendió con una solución técnicamente correcta, la que aprobó reconociendo su punto débil en la primera respuesta, y las preguntas que decidieron cada una.

Defensa 1 · Técnicamente correcta, suspendida.

DECISIÓN DEFENDIDA

«migrar el catálogo a una base no relacional para soportar el crecimiento»

EXPOSICIÓN, 10 minutos

8 minutos describiendo el modelo de datos y las ventajas
 1 minuto sobre el plan de migración
 1 minuto de conclusión

→ ninguna alternativa mencionada

→ ningún número de volumen actual

Y las preguntas que lo desmontaron:

PANEL «¿cuál es el volumen actual y cuál es el codo del sistema de ahora?»

CAND. «no lo tengo medido, pero la base relacional no escala»

→ la clase 262 dice exactamente lo contrario de esa frase: el limitante casi nunca es el motor

PANEL «¿cuál es la opción más barata y por qué no la elegiste?»

CAND. «seguir como estamos, pero no aguantaría»

→ alternativa de paja: no se presentó nada bueno de ella

PANEL «¿qué empeora con tu decisión?»

CAND. «nada relevante; ganamos escalado»

→ y aquí el panel dio la defensa por perdida

PANEL «¿cómo se hacen las consultas por rango de precio y categoría, que hoy son el 40 % del tráfico?»

CAND. (silencio) «habría que modelarlas distinto»

PANEL «¿quién opera esto de guardia y qué sabe hacer?»

CAND. «el mismo equipo»

Y la puntuación:

acotación	15	9
alternativas	20	3
criterio	20	4
compromisos	20	0
respuesta a preguntas	15	6
revisión	10	2
	■■■■■	■■■■■
	100	24

Y el comentario del panel:

«la decisión podría ser correcta. No lo sabemos, y quien la defiende tampoco, porque no midió el sistema actual ni consideró nada más. Decir que no empeora nada es lo que más nos preocupa: significa que no ha entendido la alternativa que descarta.»

Defensa 2 · Aprobada reconociendo el punto débil.

DECISIÓN DEFENDIDA

«no hacer activo-activo entre regiones y quedarnos en una región activa con otra en frío» clase 272

EXPOSICIÓN, 10 minutos

- 2 min el problema y las restricciones: 9 personas de plataforma, guardia de 6, sin requisito de residencia
- 2 min atributos ordenados con negocio, y quién los ordenó
- 3 min tres alternativas, con lo bueno de cada una
→ y de la más cara dijo: «es la única que cumple literalmente lo que negocio pidió al principio»
- 2 min el criterio: coste por hora de caída evitada, contra 31.000 USD por hora medidos por negocio
- 1 min qué empeora: 4 horas de exposición por caída de región; y qué nos obliga: ensayar la conmutación cada trimestre y mantener las cuotas de la secundaria

Y la primera pregunta, que era la hostil:

PANEL «¿y si la caída ocurre en temporada alta, cuando una hora vale cuatro veces más?»

CAND. «Entonces mi criterio se rompe. Con 124.000 por hora, activo-activo se justifica durante esas seis semanas y no durante las otras cuarenta y seis. Y

no tengo una respuesta buena para eso: no puedo montar y desmontar una región activa dos veces al año. Lo que hago es reducir la probabilidad en esa ventana: congelar cambios de alto impacto, ensayar la conmutación antes de que empiece y ampliar las cuotas de la secundaria. Es una mitigación, no una solución.»

→ el panel anotó: «reconoce el límite de su criterio en la primera respuesta»

Y dos preguntas más:

PANEL «¿qué te haría cambiar de opinión?»

CAND. «que el valor de una hora suba de 90.000 de forma sostenida, o que el equipo pase de 15 personas. Están escritas las dos en el registro, con alerta.»

PANEL «¿qué parte de esto no has probado?»

CAND. «la conmutación completa con carga real. Hemos conmutado con cuatro instancias y las cuotas de la secundaria solo daban para el 3,2 % de la capacidad. Eso lo arreglamos, pero el ensayo con carga completa está pendiente y es mi mayor riesgo abierto.»

→ el panel anotó: «dice lo que no ha probado sin que se lo saquemos»

Y la puntuación:

acotación	15	14
alternativas	20	18
criterio	20	19
compromisos	20	18
respuesta a preguntas	15	15
revisión	10	10
	■■■■■	■■■■■
	100	94

Y el cierre de esa persona, en los 5 minutos finales:

«cambiaría dos cosas tras esta conversación. Una: el criterio debería estar ponderado por temporada, no ser un número anual; lo tenía como constante y no lo es. Y dos: voy a poner fecha al ensayo de conmutación con carga completa antes de la próxima temporada alta, porque responder “está pendiente” dos veces seguidas ya es una respuesta mala.»

→ y ese tramo, que muchos desperdician, subió la nota más que ninguna respuesta

Los datos agregados de 14 defensas del equipo.

		sobre peso	por ese criterio
acotación	15	11,2	1
alternativas	20	9,4	8
criterio	20	12,1	4
compromisos	20	8,7	9
respuesta a preguntas	15	10,9	2
revisión	10	4,1	6

→ 9 de 14 fallaron en «qué empeora»

→ 8 de 14 llevaron alternativas de paja

→ y solo 5 de 14 tenían condición de revisión

Y el patrón que el panel escribió al final del ejercicio:

«casi nadie suspende por no saber. Suspenden por presentar su decisión como si no tuviera coste. Y eso, en una revisión real, no lo perdona nadie: si no dices tú lo que empeora, lo dirá otro, y entonces se pone en duda todo lo demás que has dicho.»

→ que es exactamente el error 2 de la clase 272

La lección que esta clase deja: la defensa que suspendió con 24 puntos no era técnicamente incorrecta; suspendió por responder «nada relevante» a qué empeoraba, lo que el panel leyó como no haber entendido la alternativa descartada. Y la que sacó 94 empezó reconociendo, en la primera respuesta, que su criterio se rompe en temporada alta y que no tenía una solución buena para eso, solo una mitigación.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-22-specializations-certifications-career/276-proyecto-defensa-tecnica-ante-panel/lab.py
```

El laboratorio selecciona el motor de práctica capstone y produce labresult.json. El escenario, sus comprobaciones y el artefacto esperado corresponden a esta clase; no requiere credenciales y deja explícito qué debe revalidarse en un sandbox real.

1. Lee exercise.steps y formula una predicción antes de ejecutar.
2. Ejecuta la práctica y verifica todos los elementos de checks.
3. Provoca el caso de negativetest y explica la señal observada.
4. Materializa technical-defense en evidence/ usando la plantilla indicada.
5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

El artefacto principal es un incremento integrado, demostrable y documentado. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye technical-defense para el caso CloudShop. Incluye una alternativa descartada, un supuesto que pueda falsarse, una prueba de fallo y una decisión de rollback.

■ Criterio de aceptación

- [] lab.py termina con código 0 y genera JSON válido.
- [] La entrega conecta al menos tres requisitos con mecanismos verificables.
- [] Existe una prueba positiva y una prueba negativa con evidencia.
- [] Seguridad, costo y operación aparecen como decisiones, no como anexos.
- [] Se declara una limitación y una condición que obligaría a revisar el diseño.
- [] Otra persona puede repetir el recorrido sin conocimiento tácito.

■ Errores frecuentes

Síntoma	Causa probable	Corrección
La defensa suspende con una decisión técnicamente correcta	Se presentó como si no tuviera coste y sin alternativas reales	Di qué empeora antes de que te lo pregunten y presenta lo bueno de las opciones descartadas; una alternativa de paja invalida la comparación.
El panel pregunta por volúmenes y no hay respuesta	Se decidió sin medir el sistema actual	Lleva el codo medido y el recurso limitante actual; una migración justificada por que algo no escala necesita demostrar dónde está el límite.
La conversación se convierte en una discusión personal	Se defiende la decisión en vez del criterio	Sostén el criterio; si aparece un dato nuevo que lo rompe, dilo y ajusta, que es lo que el panel evalúa.
Se desperdicia el tramo final de la defensa	Se usa para repetir la conclusión en vez de responder a lo aprendido	Cierra diciendo qué cambiarías tras la conversación y con qué plazo; ese tramo sube la nota más que ninguna respuesta.
La decisión se defiende bien y no sobrevive seis meses	No tiene condición de revisión escrita	Escribe qué supuesto, si cambia, la invalida, y pon una alerta cuando se acerque; sin eso pasa a ser herencia.
El panel pregunta qué no has probado y la respuesta improvisa	No se identificó el riesgo abierto antes de la defensa	Ten localizado y dicho tu mayor riesgo sin probar; decirlo tú da credibilidad, que te lo saquen la quita.

■ Seguridad, ética y costo

Trabaja con cuentas propias o sandboxes autorizados. No publiques secretos, identificadores reales ni datos personales. Antes de crear recursos pagos define presupuesto, etiquetas y comando de destrucción; después verifica que no queden recursos huérfanos. Los ejemplos locales enseñan contratos, pero no certifican cumplimiento ni disponibilidad de producción.

■ Preguntas de comprobación

1. ¿Qué ocho piezas hay que llevar preparadas a una defensa técnica?
2. ¿Por qué se suspende más por compromisos y alternativas que por conocimiento?
3. ¿Cuál de las cinco predicciones falló y qué confundió?
4. ¿Qué distingue la ley 31 de la ley 16?
5. ¿Cuál es la métrica que salva a una función de apoyo?

Referencias

- Nygard, M. (2011). Documenting architecture decisions.
<<https://cognitect.com/blog/2011/11/15/documenting-architecture-decisions>>
- Skelton, M. y Pais, M. (2019). Team Topologies — funciones de apoyo y su deriva.
<<https://teamtopologies.com/book>>
- Google (2024). DORA capabilities: streamlining change approval.
<<https://dora.dev/capabilities/streamlining-change-approval/>>
- AWS (2024). Well-Architected Framework Reviews.
<<https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html>>
- Ford, N. y Richards, M. (2021). Software Architecture: the hard parts — defender compromisos.
<<https://www.oreilly.com/library/view/software-architecture-the/9781492086888/>>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Anterior	Índice	Siguiente
← 275 · Portafolio, evidencia, README y entrevista de sistemas	Parte 22 · Programa	277 · Capstone retail: comercio multi-región →

Evaluación — 276 Proyecto: defensa técnica ante panel

Preguntas

1. Explica el problema que resuelve esta capacidad sin mencionar un proveedor.
2. Identifica una frontera de responsabilidad y su propietario.
3. Compara dos alternativas mediante confiabilidad, seguridad, costo y operación.
4. ¿Qué demuestra el laboratorio y qué no puede demostrar?
5. Propón un caso límite o una prueba de fallo.

Reto verificable

Entrega technical-defense con diagrama o configuración, evidencia de ejecución, alternativa descartada, riesgo residual y criterio de rollback.

Criterio de aceptación

- ☐ Resultado reproducible por una segunda persona.
- ☐ Evidencia vinculada a cada afirmación técnica.
- ☐ Prueba negativa o de fallo incluida.
- ☐ Costos y permisos expresados con unidades y alcance.
- ☐ Limitaciones declaradas sin presentar la simulación como producción.

Escala

Nivel	Evidencia
A — competente	Cumple todo y defiende trade-offs con datos.
B — en desarrollo	Cumple lo esencial; falta profundidad en una dimensión.
C — inicial	Artefacto parcial o conclusión sin evidencia suficiente.
0	Sin entrega reproducible.