Multi-Cloud Engineering Program

Parte 21 — Operación cloud, automatización y respuesta a incidentes

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	21 de 23
Clases	253–264
Evaluaciones	12
Horas estimadas	52

Índice

Parte 21 — Operación cloud, automatización y respuesta a incidentes

← Parte 20 · Índice completo · Parte 22 →

Descargar: Esta parte en PDF · Manual integral

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

Convertir la operación diaria en procesos observables, repetibles y progresivamente automatizados.

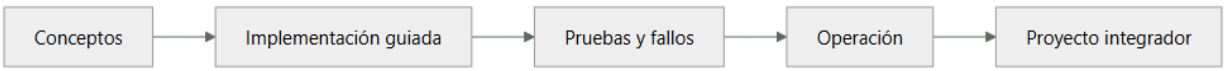
Resultados de la parte

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

Operar es controlar cambios bajo incertidumbre: inventario, señal, diagnóstico, acción reversible y aprendizaje deben formar un ciclo cerrado.

Secuencia



Clases

ID	Clase	Laboratorio	Horas
253	Inventario, etiquetado, CMDB y ownership	governance	4
254	Patching, imágenes doradas y gestión de configuración	operations	4
255	Backups, restore testing, vaults e inmutabilidad	reliability	4
256	Administración remota sin SSH permanente	security	4
257	Alertas, on-call, escalamiento y comunicación	incident	4
258	Triage de red, cómputo, datos y dependencias	incident	4
259	Runbooks ejecutables y auto-remediation	operations	4
260	Change management, ventanas y rollback	delivery	4
261	Game days, chaos engineering y aprendizaje	chaos	4
262	Capacity planning, cuotas y gestión de demanda	capacity	4
263	AIOps, automatización asistida y límites humanos	governance	4
264	Proyecto: centro de operaciones de CloudShop	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 Site Reliability Workbook — Beyer et al..
- Seeking SRE — Murphy et al..
- Effective DevOps — Davis y Daniels.

253 — Inventario, etiquetado, CMDB y ownership

Parte: 21 — Operación cloud, automatización y respuesta a incidentes Nivel: avanzado · Horas estimadas: 4
Laboratorio: governance · Estado: EXECUTABLECORE

Propósito

Saber qué hay y de quién es, que es el requisito previo de todo lo demás en operación: no se puede parchear, retirar, presupuestar ni recuperar lo que no se sabe que existe. La clase distingue inventario de configuración, explica por qué los inventarios manuales fracasan siempre, da el método para asignar dueños de verdad, y desarrolla la práctica que más resultados produce y menos se hace: retirar.

Resultados de aprendizaje

Al finalizar podrás:

1. Construir un inventario automático que no envejezca.
2. Distinguir inventario, gestión de configuración y relaciones.
3. Asignar dueños que respondan, y detectar los que no.
4. Detectar lo que existe fuera del proceso y lo que no se usa.
5. Retirar con un procedimiento seguro y medible.

Conceptos centrales

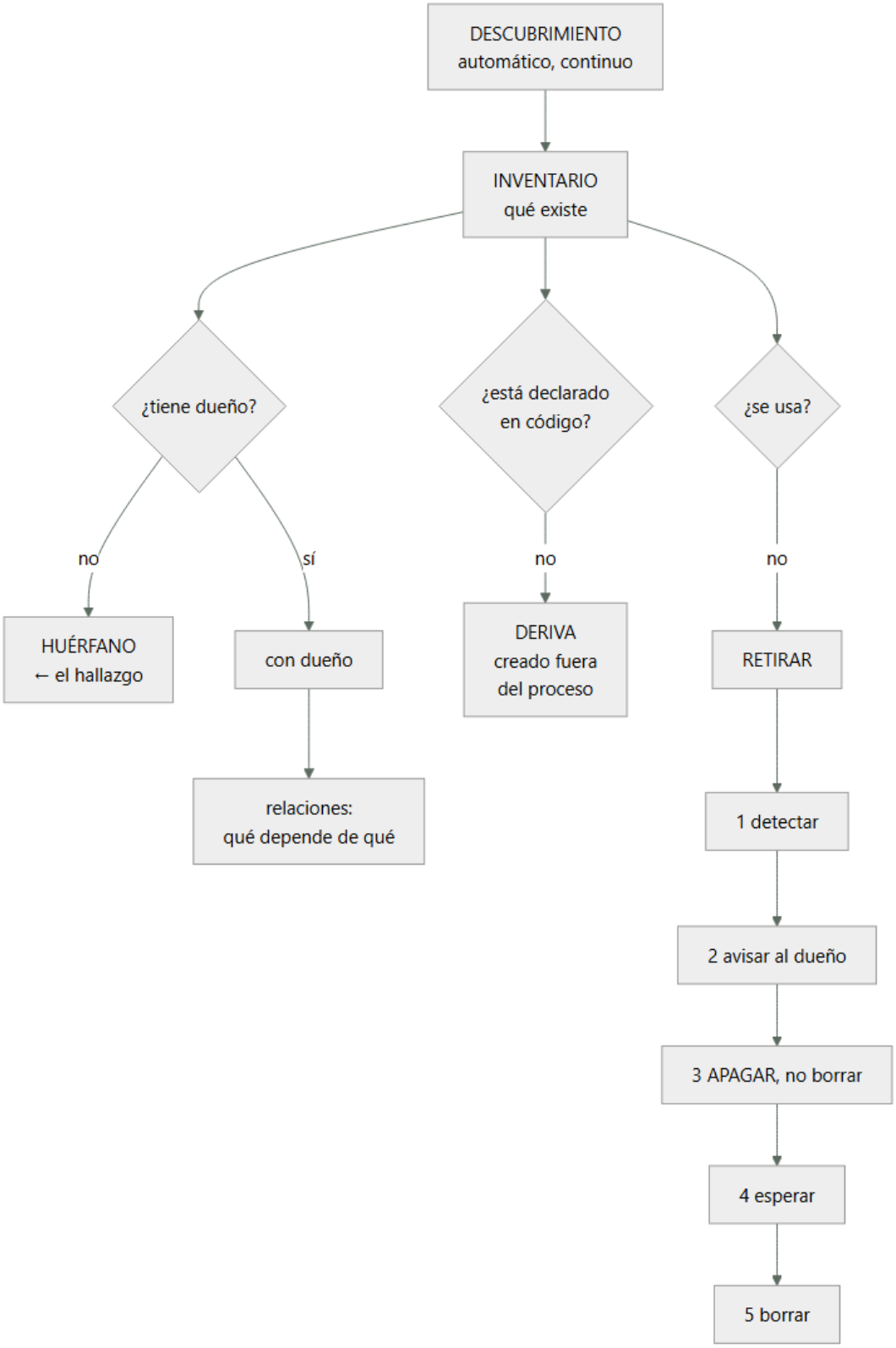
Concepto	Comprensión verificable
inventario	Lo que existe, descubierto automáticamente. La lista, no la configuración.
base de configuración	Registro de elementos, sus atributos y sus relaciones. Útil si se alimenta sola.
dueño	Equipo que responde de un recurso: lo mantiene, lo paga y decide sobre él.
recurso huérfano	El que existe sin dueño identificable. Ni se mantiene ni se retira.
deriva de inventario	Diferencia entre lo que existe y lo que está declarado o registrado.
retirada	Proceso de eliminar lo que ya no hace falta. Detectar, avisar, apagar, esperar y borrar.

Modelo mental

Operar es controlar cambios bajo incertidumbre: inventario, señal, diagnóstico, acción reversible y aprendizaje deben formar un ciclo cerrado.

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 cosas distintas

«Inventario», «base de configuración» y «mapa de dependencias» se usan como sinónimos y responden preguntas distintas.

```
INVENTARIO
  ¿QUÉ EXISTE?
```

```
  la lista, descubierta de las propias plataformas
  → y debe ser automática y continua

CONFIGURACIÓN
  ¿CÓMO ESTÁ CONFIGURADO?
  atributos, versiones, ajustes
  → y el histórico: qué cambió y cuándo

RELACIONES
  ¿QUÉ DEPENDE DE QUÉ?
  qué se cae si esto se cae
  → y esta es la que casi nunca está completa   ley 24
```

Y por qué los inventarios manuales fracasan siempre:

```
se rellenan al empezar y quedan obsoletos en semanas
nadie los actualiza porque no le sirve a quien lo
  actualizaría
y cuando hacen falta, no se pueden creer
  → y entonces alguien hace otro inventario a mano

→ el único inventario que sirve es el que se alimenta solo
→ y lo que se rellena a mano es lo que la plataforma no
  puede saber: el DUEÑO y el PROPÓSITO
```

De dónde sale el inventario automático:

```
de las API de las plataformas: recursos, cuentas, redes,
  identidades
de los clústeres y sus cargas               clase 234
de los registros de artefactos e imágenes   clase 212
de los repositorios: qué está declarado     clase 232
de los sistemas de tiquetes y de facturación
y del tráfico observado                      clase 202

→ y la unión de todo eso es lo que da la fotografía
→ con una clave común que permita cruzarlos
```

Y las relaciones, que son la parte difícil:

```
LO QUE SE PUEDE DESCUBRIR
  dependencias de red, del tráfico observado  clase 202
  llamadas entre servicios, de las trazas     clase 238
  qué consume cada dato, del linaje           clase 243
  qué imágenes usa cada carga
  y qué recursos referencia cada plantilla    clase 232

LO QUE NO
  «este proceso depende de que Ana esté disponible»
  «esta integración la mantiene un proveedor»
  y las dependencias que solo ocurren una vez al mes

→ y por eso el mapa se completa a mano donde haga falta,
  con fecha de revisión                          ley 25
```

2. El dueño, y cómo se asigna

Un recurso sin dueño no se mantiene, no se paga con criterio y no se retira. Es la ley 20 en su forma más directa.

```
QUÉ SIGNIFICA SER DUEÑO
  responder de que funcione
  pagar su coste                              clase 214
  decidir cuándo se actualiza y cuándo se retira
  y ser el destinatario de las alertas         clase 211

→ y por eso el dueño es un EQUIPO, no una persona
  → las personas cambian de equipo y se van    clase 229
```

Cómo se asigna, con lo que funciona y lo que no:

```
✗ UNA CAMPAÑA DE ETIQUETADO
  se pide a todos que etiqueten lo suyo
  → se etiqueta lo fácil y queda lo dudoso
  → y lo nuevo nace sin etiqueta                ley 27

✓ IMPONERLO EN LA CREACIÓN
```

```
  sin etiqueta de dueño, no se crea            clase 214
  → y entonces lo nuevo siempre lo tiene
  → y lo viejo se ataca aparte, con datos

Y PARA LO VIEJO, POR ORDEN
  1  quién lo creó, del registro de auditoría  clase 238
  2  quién lo despliega, del repositorio
  3  quién lo usa, del tráfico y de los permisos
  4  quién paga la cuenta o el proyecto
  5  y si nada de eso da respuesta: candidato a retirar
```

Y el criterio que resuelve los casos dudosos:

```
SI NADIE LO RECLAMA AL AVISAR, NO TIENE DUEÑO
  → y entonces se apaga
  → y si alguien se queja, ya tiene dueño

→ es brusco y es lo único que funciona
→ y por eso el paso de APAGAR antes de borrar es
  imprescindible                              clase 214
```

Los huérfanos, que son el hallazgo típico:

```
en la clase 229, de 214 proyectos
  con dueño identificable                        94
  con actividad y SIN dueño                      49
  sin actividad                                  71

en la clase 246, de 7 modelos en producción
  con dueño                                       4

y en la clase 200, un script de exportación de un empleado
que ya no estaba, copiando datos desde hacía 14 meses

→ y en todos los casos, el hallazgo lo produjo un
  inventario, no una alerta                    ley 15
```

Y la señal que dice si el modelo funciona:

```
proporción de recursos con dueño identificable
  → y su tendencia
recursos creados en el último mes sin dueño
  → debe ser cero; si no, la barrera no funciona
y tiempo desde que se detecta un huérfano hasta que se
  resuelve
```

3. Deriva: lo que existe fuera del proceso

El inventario sirve para comparar lo que hay con lo que debería haber, y esa comparación es donde están los hallazgos.

```
LAS TRES COMPARACIONES

1  INVENTARIO frente a CÓDIGO DECLARADO
   lo que existe y no está en ningún repositorio
   → creado a mano, y por tanto no reproducible
   → y no se recrea en un desastre         clases 232, 215

2  INVENTARIO frente a REGISTRO DE CONFIGURACIÓN
   lo que existe y no está registrado
   → o al revés: registrado y no existe

3  CONFIGURACIÓN ACTUAL frente a LA DECLARADA
   alguien cambió algo por la consola
   → y el siguiente despliegue lo revertirá sin avisar
                                              clase 232
```

Y lo que suele aparecer en la primera comparación:

```
recursos creados «para una prueba»              ley 25
recursos creados durante un incidente y no retirados
recursos de proyectos terminados
recursos creados por un automatismo que ya no existe
y recursos de un equipo que se reorganizó

→ y todos comparten dos propiedades: cuestan dinero y
```

```
  amplían la superficie de ataque         clases 214, 226
```

La disciplina que lo mantiene, que es la de la clase 232:

```
la comparación se ejecuta periódicamente y ALERTA
  → y no es un informe que alguien lee

y cada diferencia se clasifica
  se declara en código
  se registra como excepción, con dueño y fecha
  o se retira
→ y las tres opciones cierran; «lo miramos» no cierra
```

Y una categoría que hay que tratar aparte:

```
LO QUE LA PLATAFORMA CREA SOLA
  recursos gestionados, identidades del sistema, subredes
  de integración
  → aparecen como no declarados y son correctos
  → se marcan para ignorar, con la regla escrita
  → y sin eso, la alerta se llena de ruido y se ignora
                                              clase 125
```

Y una advertencia sobre el registro de configuración:

```
un registro de configuración que se rellena a mano tiene
el mismo destino que el inventario manual
→ se alimenta de las plataformas, o no existe
→ y su valor está en las RELACIONES y en el histórico, que
  es lo que las plataformas dan peor
```

4. Retirar

Es la práctica que más resultados produce y la que menos se hace, porque nadie la pide.

```
POR QUÉ NO SE HACE
  no hay quien la pida: ningún usuario reclama que se
  retire algo
  no tiene métrica visible
  da miedo: ¿y si alguien lo usa?
  y siempre hay algo más urgente                  ley 25

Y QUÉ CUESTA NO HACERLA
  coste directo                             clase 214
  superficie de ataque                      clase 226
  capacidad del equipo consumida en mantener  ley 23
  y actualizaciones bloqueadas por cosas que nadie usa
                                              clase 213
```

El procedimiento seguro, en cinco pasos:

```
1  DETECTAR
   sin actividad en N días
   sin consumidores, según el linaje o el tráfico
   sin dueño
   o marcado como a retirar por su dueño

2  AVISAR al dueño, si lo hay, con plazo
   → y si no lo hay, avisar en el canal general

3  APAGAR o desconectar, sin borrar
   → detener el servicio, quitar el permiso, desasociar
   → y esto es lo que hace el proceso seguro

4  ESPERAR
   → 14 o 30 días, según la criticidad
   → y si alguien se queja, se vuelve a encender en
     minutos

5  BORRAR
   → y con las copias de seguridad, aparte  clase 255
```

Y las señales de «sin actividad», por tipo:

```
instancias y contenedores    sin peticiones ni conexiones
almacenes                    sin lecturas ni escrituras
bases de datos               sin conexiones
```

```
tablas y conjuntos           sin consumidores    clase 243
identidades y claves         sin uso             clase 230
direcciones y balanceadores  sin destinos
imágenes                     sin despliegues     clase 212
suscripciones y colas        sin consumo         clase 237
reglas y rutas               sin coincidencias   clase 231
alertas                      sin disparos en un año
                                                 clase 190
y modelos                    sin peticiones      clase 246
```

Y la medida que dice si se está haciendo:

```
recursos retirados al mes
coste liberado
reclamaciones tras apagar   ← y su proporción
  → si es cero durante meses, se puede ser más agresivo
  → si es alta, la detección está mal

y la cifra que este programa ha visto varias veces
  en la clase 214, 214 recursos retirados y 3
  reclamaciones
  en la clase 229, 71 proyectos retirados y 2
```

Y el caso que hay que tratar con cuidado:

```
LO QUE SE USA UNA VEZ AL AÑO
  el cierre anual, la campaña de temporada, el informe
  regulatorio
  → «sin actividad en 90 días» lo marca como retirable
  → y apagarlo se descubre el día que hace falta

→ por eso la ventana de detección debe cubrir un ciclo
  completo de negocio, o el dueño debe poder declararlo
                                                clase 167
```

Y la lista de comprobación de la clase:

- el inventario se alimenta automáticamente de las plataformas
- cubre recursos, identidades, datos, imágenes y modelos
- hay clave común para cruzar las fuentes
- las relaciones se descubren del tráfico, las trazas y el linaje
- lo que no se puede descubrir está declarado con fecha
- el dueño es obligatorio en la creación
- el dueño es un equipo, no una persona
- hay proceso para asignar dueño a lo antiguo
- se compara el inventario con el código declarado
- se compara la configuración actual con la declarada
- lo que crea la plataforma está marcado para ignorar
- cada diferencia se declara, se exceptúa o se retira
- hay proceso de retirada con apagado antes del borrado
- la ventana de detección cubre un ciclo de negocio
- se miden retiradas, coste liberado y reclamaciones

Y el cierre que enlaza con la clase siguiente: sabiendo qué hay y de quién es, lo primero que hay que hacer con ello es mantenerlo al día. Parcheo, imágenes doradas y gestión de configuración es la materia de la clase 254.

Ejemplo trabajado

CloudShop monta su inventario. Lo que sigue es la primera fotografía, los recursos que nadie sabía que existían, y la campaña de retirada que liberó 9.400 € al mes y produjo cuatro reclamaciones.

La primera fotografía, unida de siete fuentes:

```
fuentes cruzadas
  API de las tres nubes
  clústeres y sus cargas
  registros de imágenes
  repositorios de infraestructura
  facturación
  trazas y registros de flujo
  y el sistema de tiquetes
```

```
resultado
```

```
  elementos inventariados                          41.200
    con dueño identificable                        24.100  58 %
    sin dueño                                      17.100  42 %
    declarados en código                           31.900  77 %
    creados fuera del proceso                       9.300  23 %
```

Y lo que apareció al cruzar las fuentes:

```
RECURSOS QUE FACTURAN Y NO ESTÁN EN NINGUNA OTRA FUENTE
  1.140
  → existen, se pagan, y no los despliega nadie, no los usa
    nadie y no están en ningún repositorio

RECURSOS EN REPOSITORIOS QUE NO EXISTEN
    310
  → código que declara cosas que se borraron a mano
  → y que un despliegue recrearía             clase 232

IDENTIDADES SIN USO EN 180 DÍAS
    620
  → de ellas, 41 con permisos de administración
                                                clase 230

IMÁGENES SIN DESPLIEGUE EN 1 AÑO
  7.940
  → 2,1 TB                                    clase 212

ALERTAS SIN DISPARARSE EN 1 AÑO
    214
  → cada una con su regla que mantener        clase 190

Y 3 SERVICIOS EN PRODUCCIÓN QUE NADIE CONOCÍA
  → uno de ellos, procesando pagos de un canal antiguo
    con 41 transacciones al mes
  → su dueño se fue de la empresa en 2023       ley 20
```

La asignación de dueños.

```
primer intento: campaña de etiquetado
  se pidió a los 14 equipos que etiquetaran lo suyo
  en 6 semanas
    etiquetados                               8.400
    sin etiquetar                             8.700
  → se etiquetó lo evidente y quedó lo dudoso
  → y en esas 6 semanas se crearon 1.100 recursos nuevos,
    de los cuales 340 sin etiqueta              ley 27

segundo intento: barrera primero
  1  etiqueta de dueño OBLIGATORIA en la creación, por
     política                              clase 214, 217
     → recursos nuevos sin dueño: 0
  2  y para los 8.700 antiguos, por orden de evidencia

     quién lo creó (registro de auditoría)      3.100
     quién lo despliega (repositorios)          2.400
     quién lo usa (tráfico y permisos)          1.900
     quién paga el proyecto                       900
     ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
     resueltos                                  8.300
     sin respuesta                                400
       → candidatos a retirar

tiempo                                         9 semanas
  → y casi todo automatizado: el trabajo manual fueron los
    400 dudosos
```

La campaña de retirada.

```
criterios de detección, por tipo
  instancias sin conexiones en 90 días
  almacenes sin acceso en 180 días
  bases sin conexiones en 90 días
  identidades sin uso en 180 días
  imágenes sin despliegue en 365 días
  alertas sin disparo en 365 días
```

```
  colas sin consumo en 30 días
  y proyectos sin actividad en 90 días

→ y con una excepción: los dueños pudieron marcar 41
  elementos como «de uso anual», con motivo
  → el cierre contable, el informe regulatorio y las
    campañas de temporada                    clase 167
```

Y la ejecución, en cinco pasos:

```
semana 1   DETECTAR
  candidatos                                   3.140

semana 2   AVISAR
  a los dueños, con 14 días
  reclamados por su dueño                        310
    → de ellos, 41 con motivo válido: se marcan
    → 269 no lo usaban y lo confirmaron al mirarlo
  candidatos tras el aviso                     2.830

semanas 4-6  APAGAR, sin borrar
  instancias detenidas                           410
  almacenes con acceso revocado                  180
  identidades desactivadas                       620
  colas con consumidores parados                  14
  alertas desactivadas                           214
  imágenes marcadas para caducidad             7.940
  proyectos apagados                              71
  y el resto, desasociado

semanas 6-10  ESPERAR
  reclamaciones                                    4  ←
    · un informe trimestral que usaba un almacén
    · una integración de un socio, mensual
    · un proceso de conciliación anual, no marcado
    · y una identidad de una herramienta de un proveedor
  → los 4 restaurados en menos de 15 minutos
  → y la lección: la ventana de 90 días no cubre lo
    trimestral ni lo anual                     clase 167

semana 11  BORRAR
  elementos borrados                           2.826
```

Y el resultado:

```
coste liberado                               9.400 €/mes
  imágenes y almacenamiento                    1.900 €
  instancias y bases                           5.100 €
  proyectos completos                          1.800 €
  otros                                          600 €

reducción de superficie
  identidades con permisos                      -620
  de ellas, con administración                   -41
  servicios expuestos                            -19

y las alertas
  214 retiradas
  → 214 reglas menos que mantener             ley 23
```

Y el ajuste del criterio:

```
tras las 4 reclamaciones
  la ventana de detección pasó a 400 días para almacenes y
  bases
  y a 90 días solo para instancias y contenedores
  → y los dueños pueden declarar el ciclo de uso de cada
    elemento

en la campaña siguiente (6 meses después)
  candidatos                                     810
  reclamaciones                                    1
```

Los tres servicios desconocidos:

```
1  procesamiento de pagos de un canal antiguo
   41 transacciones al mes, 3.100 € de volumen
   dueño   nadie desde 2023
   → se asignó al equipo de pagos
   → y al revisarlo: usaba una biblioteca con 3
     vulnerabilidades graves y una clave estática de 2022
                                        clases 212, 230
   → corregido

2  exportación nocturna a un socio
   → el socio ya no operaba desde 2024
   → apagado; 0 reclamaciones               ley 25

3  panel interno de un equipo disuelto
   → 4 personas lo usaban, de otro equipo
   → dueño asignado a ese equipo
```

La deriva, vigilada después:

```
comparaciones automáticas, semanales
  inventario frente a código declarado
  configuración actual frente a declarada     clase 232
  y elementos sin dueño

primeros 6 meses
  diferencias detectadas                          141
    creadas a mano durante incidentes              62
      → 48 debían haberse retirado y no se retiraron
                                                    ley 25
    creadas por automatismos de la plataforma      41
      → marcadas para ignorar, con regla escrita
    cambios por consola no revertidos              23
    y 15 sin explicación
      → investigadas: 12 eran de proveedores externos, 3
        de scripts olvidados

elementos sin dueño creados en 6 meses              0
```

El resultado, al año:

```
elementos inventariados                   41.200      38.100
recursos con dueño identificable            58 %        99 %
recursos creados fuera del proceso         9.300         310
servicios desconocidos                         3           0
identidades sin uso                          620          40
imágenes sin despliegue                     7.940         210
alertas que nunca se disparan                214           0
coste liberado                                 —      9.400 €/mes
reclamaciones tras retirar                     —        5/3.600
tiempo para saber qué hay                imposible    40 s
```

La lección que esta clase deja: la campaña de etiquetado etiquetó lo evidente y creó trescientos cuarenta recursos nuevos sin etiqueta mientras se hacía; lo que funcionó fue poner la barrera primero y atacar lo viejo con datos. Y la retirada de dos mil ochocientos veintiséis elementos produjo cuatro reclamaciones, todas de procesos trimestrales o anuales que la ventana de noventa días no podía ver: el criterio de detección tiene que cubrir un ciclo completo de negocio.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-21-cloud-operations-automation/253-inventario-etiquetado-cmdb-y-ownership/lab.py
```

El laboratorio selecciona el motor de práctica governance 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-inventory 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 jerarquía de política, ownership y evidencia. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye cloud-inventory 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 inventario está obsoleto a las pocas semanas	Se rellena a mano y a nadie le sirve mantenerlo	Aliméntalo automáticamente de las plataformas; solo el dueño y el propósito se declaran a mano.
La campaña de etiquetado no termina nunca	Se etiqueta lo antiguo mientras lo nuevo nace sin etiqueta	Impón la etiqueta de dueño en la creación primero y ataca lo antiguo después, con datos de auditoría, despliegue, uso y facturación.
Hay recursos que facturan y nadie reclama	Son huérfanos y ningún proceso los detecta	Cruza facturación con inventario, repositorios y tráfico; lo que aparece solo en la factura es el hallazgo.
Retirar algo rompe un proceso trimestral	La ventana de detección no cubre un ciclo completo de negocio	Ajusta la ventana por tipo de recurso y permite al dueño declarar el ciclo de uso.
La alerta de deriva se llena de ruido y se ignora	Los recursos que crea la plataforma aparecen como no declarados	Márcalos para ignorar con una regla escrita y revisa esa regla periódicamente.
Nadie retira nada aunque haya lista	Ningún usuario lo pide y siempre hay algo más urgente	Automatiza el proceso completo con apagado antes de borrar, y mide retiradas, coste liberado y reclamaciones.

■ 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 pregunta responde cada una de las tres cosas: inventario, configuración y relaciones?
2. ¿Por qué fracasan los inventarios manuales y qué es lo único que se declara a mano?
3. ¿En qué orden se busca el dueño de un recurso antiguo?

4. ¿Qué tres comparaciones producen los hallazgos de deriva?
5. ¿Cuáles son los cinco pasos de una retirada segura y cuál es el que la hace segura?

Referencias

- AWS (2025). AWS Config and Resource Explorer. <https://docs.aws.amazon.com/config/latest/developerguide/WhatIsConfig.html>
- Microsoft (2025). Azure Resource Graph. <https://learn.microsoft.com/en-us/azure/governance/resource-graph/overview>
- Google Cloud (2025). Cloud Asset Inventory. <https://cloud.google.com/asset-inventory/docs/overview>
- Limoncelli, T. y otros (2016). The Practice of Cloud System Administration. <https://the-cloud-book.com/>
- Beyer, B. y otros (2018). The Site Reliability Workbook, cap. sobre gestión de configuración. <https://sre.google/workbook/table-of-contents/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 21 en PDF · Manual integral

Anterior	Índice	Siguiente
← 252 · Proyecto: asistente operativo de CloudShop	Parte 21 · Programa	254 · Patching, imágenes doradas y gestión de configuración →

Evaluación — 253 Inventario, etiquetado, CMDB y ownership

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-inventory 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.

254 — Patching, imágenes doradas y gestión de configuración

Parte: 21 — Operación cloud, automatización y respuesta a incidentes Nivel: avanzado · Horas estimadas: 4
Laboratorio: operations · Estado: EXECUTABLECORE

Propósito

Mantener al día lo que existe, que es donde se acumula la mayor parte del riesgo operativo y del trabajo repetitivo. La clase explica por qué el parcheo tradicional fracasa y qué lo sustituye —sustituir en vez de arreglar—, cómo se construyen imágenes base que no envejecen, y qué hacer con lo que no se puede sustituir: la gestión de configuración, con su alcance real y sus límites.

Resultados de aprendizaje

Al finalizar podrás:

1. Sustituir en lugar de parchear donde el sistema lo permita.
2. Construir y renovar imágenes base con procedencia y verificación.
3. Priorizar vulnerabilidades por alcance y por explotación real.
4. Aplicar gestión de configuración donde la sustitución no llega.
5. Medir la antigüedad de lo desplegado y actuar sobre ella.

Conceptos centrales

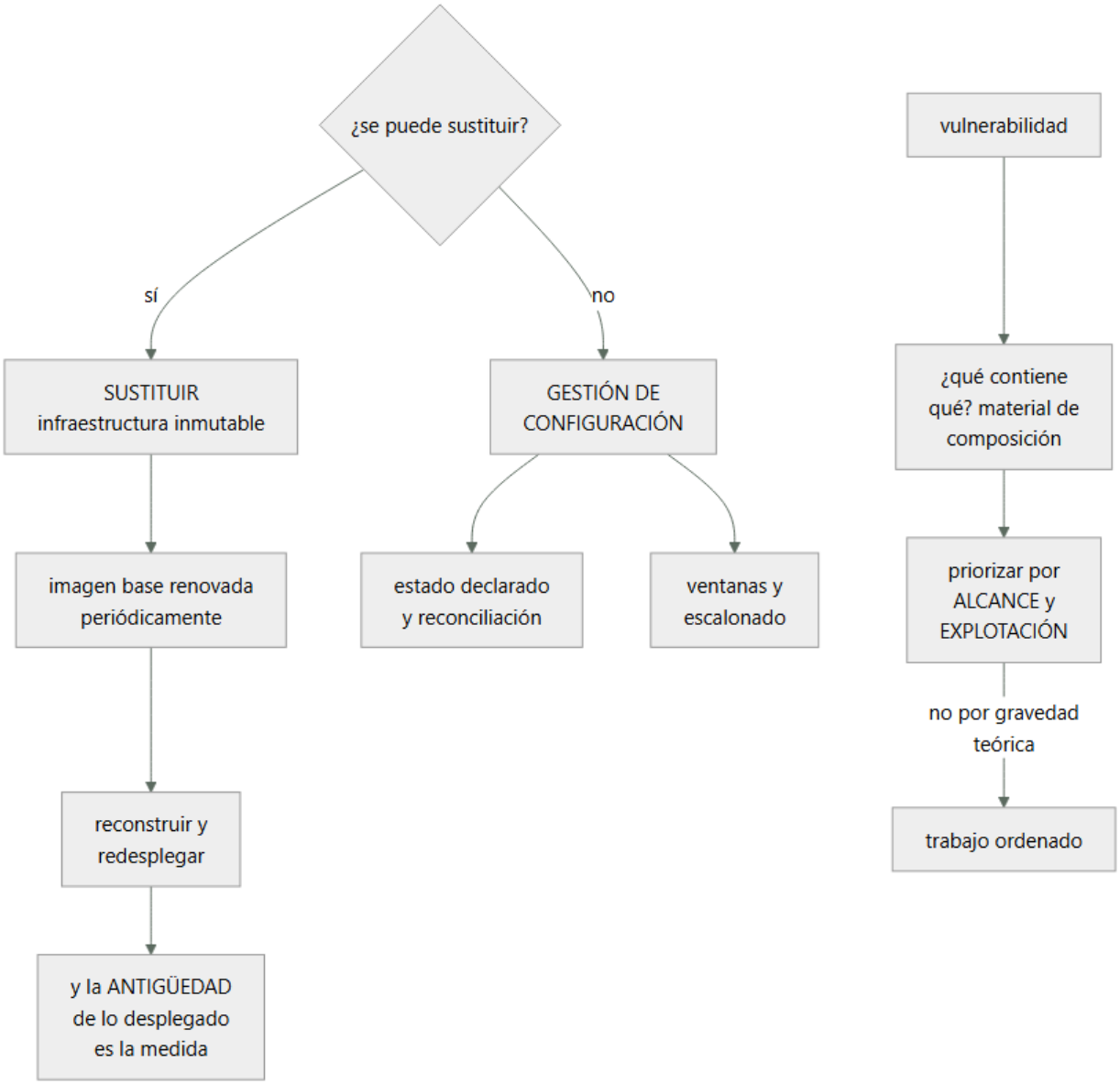
Concepto	Comprensión verificable
infraestructura inmutable	No se modifica lo desplegado: se sustituye por una versión nueva.
imagen base	Punto de partida común de las cargas. Su renovación es lo que aplica los parches a todo.
antigüedad de lo desplegado	Tiempo desde que se construyó lo que está corriendo. La medida que resume el estado del parcheo.
material de composición	Lista de lo que contiene un artefacto. Permite saber qué está afectado por una vulnerabilidad.
explotación conocida	Que exista un ataque real usando esa vulnerabilidad. Cambia la prioridad más que la gravedad teórica.
gestión de configuración	Aplicar y mantener un estado declarado sobre sistemas que no se sustituyen.

Modelo mental

Operar es controlar cambios bajo incertidumbre: inventario, señal, diagnóstico, acción reversible y aprendizaje deben formar un ciclo cerrado.

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. Sustituir en vez de arreglar

El parcheo tradicional —conectarse a un sistema y actualizarlo— falla por razones estructurales.

```
POR QUÉ FALLA
  cada sistema acaba en un estado distinto
    → «funciona en el servidor 3 y no en el 7»
  un parche que rompe algo deja el sistema a medias
  el que estaba apagado no se parchea
  el que se creó ayer parte de una imagen vieja
  y nadie sabe qué versión hay dónde

→ y el resultado es un conjunto de sistemas únicos que
  nadie puede reproducir                          clase 253
```

La alternativa, que es la que este programa usa desde la parte 05:

```
NO SE MODIFICA LO DESPLEGADO: SE SUSTITUYE
  la imagen base se renueva
  las cargas se reconstruyen sobre ella
  y se despliegan escalonadamente          clase 102

→ y así el parcheo deja de ser una operación aparte y pasa
  a ser un despliegue más
```

```
→ con su reversión, sus comprobaciones y su escalonado
```

Y lo que hace falta para que funcione:

```
1  QUE NADA IMPORTANTE VIVA EN EL SISTEMA
   estado en almacenes y bases, no en discos locales
   → y si algo tiene estado local, no se puede sustituir
     sin más                                  clase 149

2  QUE RECONSTRUIR SEA BARATO Y AUTOMÁTICO
   → si construir una imagen tarda dos horas y se hace a
     mano, no se renovará                      ley 16

3  QUE EL DESPLIEGUE SEA SEGURO
   escalonado, con comprobaciones y reversión
                                         clases 212, 233

4  Y QUE LA RENOVACIÓN ESTÉ PROGRAMADA
   → «cuando haya una vulnerabilidad» significa nunca
   → cada dos o cuatro semanas, pase lo que pase
```

La medida que resume todo esto:

```
ANTIGÜEDAD DE LO DESPLEGADO
  ¿cuánto hace que se construyó lo que está corriendo?

  → si el p95 es de 15 días, el parcheo funciona
  → si es de 8 meses, no funciona, digan lo que digan los
    informes de cumplimiento

y es mejor medida que «porcentaje de parches aplicados»
  → porque no depende de qué se considere un parche
  → y porque es una sola cifra que todo el mundo entiende
                                                  ley 17
```

Y la alerta que corresponde:

```
«esta carga lleva más de N días sin reconstruirse»
  → y es una alerta de antigüedad, como las de la
    clase 211                                  ley 13
→ y cubre lo que ninguna alerta de error cubre: lo que
  funciona y está viejo
```

2. Imágenes base y procedencia

La imagen base es el punto donde se aplica el parcheo a todo lo que la usa.

```
UNA IMAGEN BASE POR FAMILIA, no una por equipo
  → si cada equipo mantiene la suya, hay veinte imágenes
    que renovar                                ley 23
  → y la mitad no se renuevan

Y LO QUE DEBE TRAER
  el sistema mínimo, sin herramientas de más
  las bibliotecas comunes, con versión fija
  los agentes obligatorios: telemetría, seguridad
  la configuración base endurecida
  y NADA de secretos                      clase 212
```

Y el ciclo de renovación:

```
1  construcción PROGRAMADA, cada 2-4 semanas
2  exploración de vulnerabilidades
3  pruebas: que arranca, que los agentes reportan, que
   pasan las comprobaciones
4  publicación con ETIQUETA INMUTABLE       clase 212
5  y las cargas la adoptan en su siguiente despliegue

→ y aquí hay una decisión
  ¿las cargas se redespliegan solas al haber imagen nueva,
   o esperan a su siguiente cambio?
  → si esperan, una carga que no cambia nunca se queda
    vieja                                     ley 25
  → lo razonable: redespliegue automático en entornos
    inferiores, y programado en producción
```

La procedencia, que es lo que permite responder a una vulnerabilidad:

```
MATERIAL DE COMPOSICIÓN
  qué contiene cada artefacto: paquetes, bibliotecas,
  versiones
  generado al construir, no a posteriori

→ y así, ante una vulnerabilidad nueva, la pregunta «¿qué
  está afectado?» se responde con una consulta
→ sin él, se responde revisando a mano y se tarda días

Y LA FIRMA
  el artefacto se firma al construir y se verifica al
  desplegar                                  clase 106
  → y la admisión rechaza lo no firmado     clase 234
```

Y el registro de imágenes, con su higiene:

```
etiquetas inmutables
exploración CONTINUA, no solo al subir
  → una imagen desplegada hace tres meses puede tener hoy
    una vulnerabilidad conocida                clase 212
caducidad: las viejas se borran
y alerta: «hay imágenes en producción con vulnerabilidades
  graves publicadas después de su construcción»
```

Y una decisión que se hace mal:

```
✗ RECONSTRUIR SOLO CUANDO HAY VULNERABILIDAD
  → convierte el parcheo en una urgencia cada vez
  → y la urgencia se hace mal

✓ RECONSTRUIR SIEMPRE, EN CALENDARIO
  → y entonces una vulnerabilidad es «adelantamos la
    siguiente construcción», que es un procedimiento
    conocido                                    ley 22
```

3. Priorizar vulnerabilidades

Una exploración produce cientos de hallazgos. Ordenarlos por gravedad teórica es la forma más rápida de trabajar mucho y reducir poco riesgo.

```
LO QUE ORDENA DE VERDAD, en este orden

1  ¿HAY EXPLOTACIÓN CONOCIDA?
   ¿existe un ataque real usando esto?
   → hay catálogos públicos de vulnerabilidades explotadas
   → y esas van primero, sea cual sea su gravedad teórica

2  ¿ES ALCANZABLE?
   ¿el componente vulnerable está expuesto?
   ¿el código afectado se ejecuta siquiera?
   → una biblioteca incluida y no usada tiene riesgo casi
     nulo

3  ¿HASTA DÓNDE SE LLEGA DESDE AHÍ?
   el alcance desde el punto comprometido
                                       clases 189, 226

4  ¿CUÁNTOS ARTEFACTOS AFECTA?
   → y aquí el material de composición da la respuesta

5  Y SOLO ENTONCES, la gravedad teórica
```

Y la consecuencia práctica:

```
de 400 hallazgos «críticos» de una exploración típica
  con explotación conocida              entre 5 y 20
  alcanzables de verdad                 menos de la mitad
→ y esos son el trabajo

→ perseguir los 400 consume el trimestre y no reduce el
  riesgo tanto como resolver los 12 primeros
                                                clase 226
```

Y lo que se hace con el resto:

```
se resuelve en el ciclo normal de renovación
  → la mayoría desaparece sola al reconstruir
y lo que no se puede arreglar se acepta por escrito, con
  dueño y fecha                              clase 189
```

La cadena de suministro, que es la parte que más ha crecido:

```
las dependencias de terceros son la mayor superficie
  → y una dependencia trae otras: el árbol es enorme

lo que hay que hacer
  fijar las versiones, con fichero de bloqueo
  verificar la procedencia de lo que se descarga
  usar un repositorio propio que actúe de espejo
    → así una dependencia retirada del origen no rompe la
      construcción                                ley 25
  y comprobar que un paquete nuevo no es un nombre
    parecido a otro

y lo que no hay que hacer
  actualizar todo automáticamente sin pruebas
  → una actualización maliciosa entra igual que una buena
                                               clase 106
```

Y una advertencia sobre las herramientas:

```
las exploraciones producen FALSOS POSITIVOS
  versiones que el proveedor ha corregido sin cambiar el
  número
  componentes presentes y no usados
→ y si el equipo pierde la confianza en la herramienta,
  deja de mirarla                                ley 15
→ por eso las exclusiones justificadas son parte del
  trabajo, no una trampa
```

4. Lo que no se puede sustituir

Siempre queda algo: bases de datos, sistemas heredados, dispositivos, equipos de red.

```
DÓNDE NO LLEGA LA SUSTITUCIÓN
  bases de datos con estado grande
    → se actualizan en su sitio, con su procedimiento
  sistemas heredados que no se pueden reconstruir
  dispositivos y equipos de red             clase 203
  y estaciones de trabajo

→ y ahí sigue haciendo falta gestión de configuración
```

Cómo se hace bien:

```
ESTADO DECLARADO Y RECONCILIACIÓN
  se declara cómo debe estar el sistema
  un agente lo compara y corrige la diferencia
  → y la deriva se detecta                   clase 253

EJECUCIÓN ESCALONADA
  nunca a todos a la vez                       ley 25
  → un grupo, comprobar, siguiente grupo
  → y con criterio de parada

VENTANAS DE MANTENIMIENTO
  declaradas, y con exclusiones para campañas
                                          clases 222, 234

Y COMPROBACIÓN POSTERIOR
  ¿el sistema sigue funcionando tras el cambio?
  → y no solo «el paquete se instaló»
```

Y el orden de una actualización de base de datos, que es el caso más delicado:

```
1  comprobar compatibilidad de la aplicación
2  ensayar en una copia restaurada de producción
   → y medir cuánto tarda                    clase 255
3  ventana declarada, con el negocio avisado
```

```
4  copia inmediatamente antes
5  actualizar la réplica primero, si la hay
6  conmutar, comprobar y solo entonces la primaria
7  y vuelta atrás preparada y PROBADA          ley 22
```

La antigüedad, medida en todo:

```
cargas: días desde la construcción
imágenes base: días desde la renovación
bases de datos: versión frente a la soportada
dispositivos: versión de firmware
y dependencias: días desde la última actualización

→ y una sola tabla con esas cifras dice más del estado del
  parcheo que cualquier informe de cumplimiento
                                              clase 226
```

Y las señales que hay que vigilar:

```
p50 y p95 de antigüedad de lo desplegado
cargas por encima del umbral de antigüedad
vulnerabilidades con explotación conocida, abiertas
imágenes en producción con vulnerabilidades publicadas
  después de su construcción
sistemas fuera de soporte
y fallos de reconstrucción programada
  → si la construcción periódica falla y nadie lo mira, el
    parcheo se ha parado en silencio          ley 13
```

Y la lista de comprobación de la clase:

- las cargas se sustituyen, no se parchean en el sitio
- nada importante vive en el disco local
- la reconstrucción está automatizada y es barata
- hay imagen base por familia, no por equipo
- la imagen se renueva en calendario, no por urgencia
- se genera material de composición al construir
- los artefactos se firman y se verifican al desplegar
- hay exploración continua, no solo al construir
- las vulnerabilidades se priorizan por explotación y alcance
- las exclusiones están justificadas y revisadas
- las dependencias están fijadas y con espejo propio
- lo que no se sustituye tiene estado declarado y reconciliación
- las actualizaciones son escalonadas y con ventana
- la vuelta atrás está probada
- se mide la antigüedad de lo desplegado, con alerta
- hay alerta si falla la reconstrucción programada

Y el cierre que enlaza con la clase siguiente: mantener al día lo que existe protege de lo previsible. Lo imprevisible exige poder volver atrás, y para eso hacen falta copias que funcionen. Es la materia de la clase 255.

Ejemplo trabajado

CloudShop rehace su parcheo. Lo que sigue es la antigüedad de lo desplegado que nadie había medido, la priorización que redujo 412 hallazgos críticos a 14, y la reconstrucción programada que llevaba tres meses fallando en silencio.

La primera medición.

```
el informe de cumplimiento decía
  «97 % de sistemas parcheados»

y la antigüedad de lo desplegado, medida
  cargas en contenedores                    310
    p50                                   9 días
    p95                                 214 días  ←
    máximo                              441 días
  máquinas virtuales                         88
    p50                                 118 días
    p95                                 390 días
  imágenes base en uso                        7
    la más reciente                      14 días
    la más antigua                      290 días  ←
```

```
→ el 97 % se refería a que el agente de parcheo había
  ejecutado con éxito
→ y ejecutaba sobre las máquinas encendidas en ese momento
→ las cargas en contenedores no las tocaba en absoluto
```

Y el desglose de las cargas más antiguas:

```
las 41 cargas con más de 180 días
  servicios que no habían cambiado desde su despliegue     28
    → funcionaban, nadie los tocaba, nadie los reconstruía
                                                      ley 25
  servicios de equipos disueltos                          7
  trabajos programados                                    6

→ y las 28 primeras son el caso puro: si la reconstrucción
  depende de que alguien cambie el código, lo que no cambia
  se queda viejo
```

La reconstrucción programada.

```
lo que se montó
  las 7 imágenes base se reducen a 3 (una por familia)
  construcción programada cada 14 días
  exploración, pruebas de arranque y de agentes
  publicación con etiqueta inmutable
  y REDESPLIEGUE AUTOMÁTICO de todas las cargas
    en desarrollo y preproducción, inmediato
    en producción, escalonado durante la semana siguiente
                                              clase 212

y la decisión que importaba
  el redespliegue NO espera a que el equipo cambie algo
  → lo que no cambia se reconstruye igual
```

Y el resultado:

```
p50 de antigüedad (contenedores)          9 días      6 días
p95                                     214 días     19 días
máximo                                  441 días     24 días
cargas por encima de 30 días                 41           0
```

La reconstrucción que fallaba en silencio.

```
tres meses después, la p95 volvió a subir a 74 días

diagnóstico
  la construcción programada de una de las tres imágenes
  fallaba desde hacía 11 semanas
  motivo: un repositorio de paquetes había cambiado su
  dirección                                  clase 195

  y el fallo
    la canalización marcaba el trabajo como fallido
    la alerta existía
    y salía al canal del equipo de plataforma, que tenía
    340 mensajes al día                    ley 15, clase 211

→ y como la imagen anterior seguía existiendo, las cargas
  se seguían desplegando con ella
→ nada falló: simplemente dejó de renovarse       ley 13

correcciones
  alerta de ANTIGÜEDAD de la imagen base
    «la imagen X no se renueva desde hace más de 21 días»
    → a canal con guardia
  y espejo propio de los repositorios de paquetes
    → un cambio de dirección del origen deja de romper la
      construcción                              ley 25
```

La priorización de vulnerabilidades.

```
la exploración de las 3 imágenes y las 310 cargas
  hallazgos totales                              4.180
  clasificados como críticos o graves              412

y el equipo llevaba 5 meses persiguiendo los 412
```

```
  resueltos                                          180
  y aparecían nuevos cada semana

la priorización, aplicada
  1  con explotación conocida (catálogo público)     14  ←
  2  alcanzables desde internet                       61
     de ellos, con explotación conocida                9
  3  en componentes que el código NO usa             190
     → riesgo casi nulo; se resuelven al reconstruir
  4  en herramientas de construcción, no en
     producción                                       88
  5  el resto                                         59

→ los 14 primeros se resolvieron en 6 días
→ de los 412, 340 desaparecieron solos en las dos
  siguientes reconstrucciones programadas

y lo que quedó
  vulnerabilidades sin corrección disponible          8
  → aceptadas por escrito, con dueño, mitigación y fecha
    de revisión                                 clase 189
```

Y el material de composición, que hizo posible todo esto:

```
antes de tenerlo
  ante una vulnerabilidad nueva, la pregunta «¿qué está
  afectado?» se respondía revisando a mano
  tiempo medio                                    3 días

con el material generado al construir
  consulta sobre el inventario de artefactos  clase 253
  tiempo medio                                     40 s

y en la primera vulnerabilidad grave tras montarlo
  artefactos afectados, según la consulta            61
  que el equipo creía                                12
  → 49 que no se habrían corregido
```

Lo que no se pudo sustituir.

```
quedaron fuera de la sustitución
  4 bases de datos gestionadas
  1 sistema heredado de facturación
  11 equipos de red de las oficinas
  340 puntos de venta de las tiendas          clase 203

y para cada uno
  BASES DE DATOS
    actualización con el procedimiento de 7 pasos
    ensayada sobre copia restaurada           clase 255
    → y en el primer ensayo, la actualización tardó 4 h 10
      frente a la 1 h prevista
    → la ventana se planificó con la cifra medida

  SISTEMA HEREDADO
    gestión de configuración con estado declarado
    reconciliación diaria
    → y la deriva detectada: 14 cambios manuales en 6 meses
                                                clase 253

  EQUIPOS DE RED Y PUNTOS DE VENTA
    actualización escalonada por grupos       clase 203
    con doble partición y vuelta atrás automática
    → y en junio, una versión defectuosa se detectó en el
      grupo de 5 y no llegó a los otros 335
```

Las dependencias y la cadena de suministro:

```
ficheros de bloqueo en los 41 repositorios
  antes                                              18
  después                                            41

espejo propio de los repositorios de paquetes
  → y en 6 meses, 3 paquetes desaparecieron del origen
  → las construcciones siguieron funcionando  ley 25
```

```
y la comprobación de nombres parecidos
  al añadir una dependencia nueva, se comprueba si existe
  otra con nombre casi idéntico y mucha más descarga
  → 2 avisos en 6 meses; los 2, errores de escritura
```

El resultado, al año:

```
p95 de antigüedad (contenedores)        214 días     19 días
p95 (máquinas virtuales)                390 días     26 días
imágenes base                                  7          3
la más antigua en uso                   290 días     16 días
vulnerabilidades con explotación conocida
  abiertas                                    14          0
hallazgos perseguidos sin criterio           412          0
tiempo para saber qué está afectado        3 días       40 s
reconstrucción programada parada sin
  detectar                                 11 semanas       0
sistemas fuera de soporte                      6          0
```

La lección que esta clase deja: el informe decía 97 % de sistemas parcheados y la antigüedad real del percentil 95 era de doscientos catorce días, porque la medida contaba ejecuciones del agente y no cubría los contenedores. La reconstrucción programada, que era la solución, llevó once semanas parada sin que nadie lo notara, porque su fallo era un mensaje más entre trescientos cuarenta al día. Y de cuatrocientos doce hallazgos críticos, catorce tenían explotación conocida y trescientos cuarenta desaparecieron solos al reconstruir.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-21-cloud-operations-automation/254-patching-imagenes-doradas-y-gestion-de-configuracion/lab.py
```

El laboratorio selecciona el motor de práctica operations 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 patch-strategy 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 runbook probado por otra persona. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye patch-strategy 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 informe dice que casi todo está parcheado y no lo está	La medida cuenta ejecuciones del agente y no cubre lo que no es una máquina	Mide la antigüedad de lo desplegado, con percentiles, y alerta sobre las cargas que la superan.
Los servicios que nadie toca se quedan viejos	La reconstrucción depende de que alguien cambie el código	Redespliega automáticamente al renovar la imagen base, sin esperar a un cambio del equipo.
El parcheo se detiene sin que nadie lo note	La construcción programada falla y su alerta se pierde entre el ruido	Alerta por antigüedad de la imagen base a un canal con guardia, no solo por fallo de la construcción.
El equipo persigue cientos de hallazgos y el riesgo no baja	Se prioriza por gravedad teórica	Ordena por explotación conocida, alcanzabilidad y alcance; el resto desaparece con la renovación periódica.
No se sabe qué artefactos contienen un componente vulnerable	No se genera material de composición al construir	Genera la lista de contenido en la construcción y consúltala contra el inventario de artefactos.
Una actualización de base de datos se alarga mucho más de lo previsto	No se ensayó sobre una copia restaurada de producción	Ensaya y cronometra sobre una restauración real antes de planificar la ventana.

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 falla el parcheo tradicional y qué lo sustituye?
2. ¿Qué medida resume el estado del parcheo mejor que el porcentaje de parches aplicados?
3. ¿Qué decide si una carga que nadie toca se mantiene al día?
4. ¿En qué orden se priorizan las vulnerabilidades?
5. ¿Qué queda fuera de la sustitución y cómo se trata?

Referencias

- CISA (2025). Known Exploited Vulnerabilities catalog. <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>
- CycloneDX (2025). Software bill of materials specification. <https://cyclonedx.org/specification/overview/>
- SLSA (2025). Supply chain levels for software artifacts. <https://slsa.dev/>
- AWS (2025). EC2 Image Builder and golden AMI pipelines. <https://docs.aws.amazon.com/imagebuilder/latest/userguide/what-is-image-builder.html>
- Beyer, B. y otros (2018). The Site Reliability Workbook, cap. sobre gestión de cambios. <https://sre.google/workbook/table-of-contents/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 21 en PDF · Manual integral

Anterior	Índice	Siguiente
← 253 · Inventario, etiquetado, CMDB y ownership	Parte 21 · Programa	255 · Backups, restore testing, vaults e inmutabilidad →

Evaluación — 254 Patching, imágenes doradas y gestión de configuración

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 patch-strategy 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.

255 — Backups, restore testing, vaults e inmutabilidad

Parte: 21 — Operación cloud, automatización y respuesta a incidentes Nivel: avanzado · Horas estimadas: 4
Laboratorio: reliability · Estado: EXECUTABLECORE

Propósito

Poder volver atrás cuando algo se pierde, que es distinto de tener copias: una copia que nunca se ha restaurado no es una copia, es un fichero. La clase cubre qué hay que copiar y qué no, la inmutabilidad y el aislamiento que protegen de un borrado deliberado, el ensayo de restauración como práctica obligatoria, y los escenarios que las copias no cubren.

Resultados de aprendizaje

Al finalizar podrás:

1. Decidir qué se copia, con qué frecuencia y cuánto se conserva.
2. Proteger las copias de un borrado deliberado o accidental.
3. Ensayar restauraciones y medir el plazo real.
4. Distinguir copia, réplica y versión, y saber qué cubre cada una.
5. Reconocer los escenarios que las copias no resuelven.

Conceptos centrales

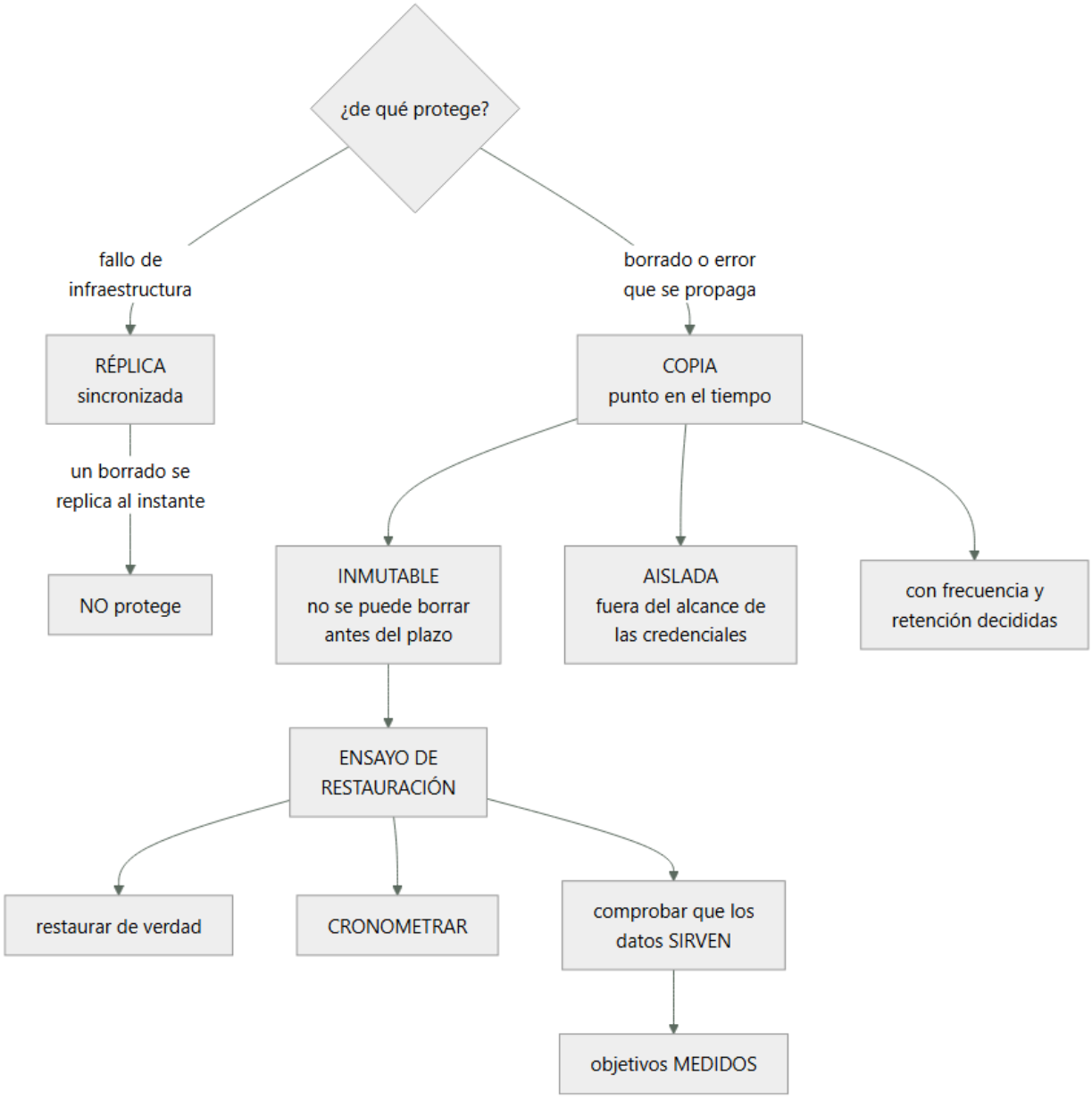
Concepto	Comprensión verificable
copia de seguridad	Punto en el tiempo del que se puede restaurar. Protege de errores y de borrados.
réplica	Copia sincronizada de los datos. Protege de fallo de infraestructura, no de borrado.
inmutabilidad	Propiedad que impide modificar o borrar una copia antes de un plazo, ni siquiera con permisos totales.
aislamiento de la copia	Que las copias vivan fuera del alcance de las credenciales del sistema copiado.
objetivo de recuperación	Cuánto se tarda en volver y cuánto se pierde. Ambos MEDIDOS, no declarados.
ensayo de restauración	Restaurar de verdad y comprobar que los datos sirven. Lo único que valida una copia.

Modelo mental

Operar es controlar cambios bajo incertidumbre: inventario, señal, diagnóstico, acción reversible y aprendizaje deben formar un ciclo cerrado.

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. Copia, réplica y versión

Tres cosas que se confunden y protegen de amenazas distintas. Confundirlas es el error inicial.

```
RÉPLICA
  copia sincronizada, en otra zona o región
  protege de   fallo de infraestructura
  NO protege de
    un borrado: se replica al instante
    una corrupción: se replica
    un error de aplicación: se replica
    ni un cifrado malicioso: se replica

COPIA DE SEGURIDAD
  punto en el tiempo del que se puede volver
  protege de   borrado, corrupción, error y ataque
  → y es lo único que protege de esas cuatro

VERSIONADO
  las versiones anteriores de un objeto se conservan
  protege de   sobrescritura y borrado accidental
```

```
  NO protege de
    quien tenga permiso para borrar las versiones
  → y por eso el versionado sin bloqueo no basta
```

Y el error que este programa ha visto varias veces:

```
«tenemos replicación multirregión, estamos cubiertos»
  → ante un borrado, las dos regiones quedan vacías a la
    vez                                          clase 166
→ y esa frase es la señal de que no hay copias de verdad
```

Qué se copia, con lo que se olvida:

```
LO QUE SE COPIA SIEMPRE
  bases de datos
  almacenes de objetos con datos de negocio
  configuración que no está en código

LO QUE SE OLVIDA
  el ESTADO del código como infraestructura   clase 232
    → sin él, el código no sabe qué gestiona
  los secretos y las claves de cifrado
    → sin la clave, los datos cifrados no sirven
                                                clase 239
  los artefactos e imágenes                    clase 254
    → sin ellos, no se puede desplegar
  el registro de decisiones y la documentación
  las definiciones de alertas, paneles y objetivos
  y los datos de los sistemas gestionados que se creen
    copiados y no lo están

LO QUE NO SE COPIA
  lo que se puede reconstruir del código          clase 232
  cachés y datos derivados
  y registros más allá de su retención        clase 238
  → copiarlos cuesta y no aporta
```

Y la pregunta que ordena las decisiones:

```
POR CADA CONJUNTO
  ¿cuánto se puede perder?     → frecuencia de la copia
  ¿cuánto se puede tardar?     → tipo de copia y de destino
  ¿cuánto hay que conservar?   → retención, por norma o por
                                 negocio
  ¿desde dónde hay que poder
   restaurar?                  → región y aislamiento

→ y las respuestas se escriben, por conjunto  clase 166
→ «lo copiamos todo cada noche» no es una respuesta
```

2. Proteger las copias

Una copia que se puede borrar con las mismas credenciales que borran los datos no protege del escenario que importa.

```
EL ESCENARIO
  alguien obtiene credenciales con permisos amplios
  borra los datos
  y borra las copias
  → y ese es el orden que sigue un ataque de cifrado

→ y también es lo que hace un error: un despliegue en modo
  completo sobre el grupo equivocado          clase 220
```

Las tres protecciones, que se aplican juntas:

```
1  INMUTABILIDAD
   la copia no se puede modificar ni borrar antes de un
   plazo
   ni siquiera con permisos totales
   → y ese plazo se elige por escenario, no por costumbre

2  AISLAMIENTO
   las copias viven en otra cuenta, otro proyecto u otra
   suscripción
   con credenciales distintas
```

```
   → y el sistema copiado NO puede escribir ni borrar allí
   → el flujo es de una sola dirección

3  SEPARACIÓN DE PERMISOS
   quien administra el sistema no administra las copias
   quien puede borrar una copia es otro, y con elevación
   temporal                              clases 218, 230
```

Y la comprobación que valida las tres:

```
desde las credenciales de producción, intentar
  borrar una copia            → debe fallar
  modificarla                 → debe fallar
  cambiar su retención        → debe fallar
  desactivar la programación  → debe fallar

→ y esta prueba negativa ha aparecido en las clases 174,
  179 y 189, y ha fallado más de una vez        ley 22
```

Y la copia fuera de línea, para el peor caso:

```
una copia periódica en un medio o una cuenta sin
conectividad con el resto
  → protege del escenario en que el atacante tiene tiempo
  → y su restauración es lenta, a propósito

→ hace falta si el sistema es crítico y el escenario de
  ataque es real
→ y si no se hace, se declara como riesgo aceptado
                                                clase 189
```

El cifrado de las copias, con su detalle:

```
las copias van cifradas
  → y la CLAVE tiene que estar disponible al restaurar
  → si la clave vive solo en la región perdida, la copia no
    sirve                                  clase 239

y la clave también se copia, o se replica
  → con su propio aislamiento
  → y esto es lo que más se olvida
```

Y la retención, con lo que hay que decidir:

```
¿CUÁNTO ATRÁS HAY QUE PODER VOLVER?
  un borrado se detecta en horas
  una corrupción, en días
  un error de lógica, en semanas o meses
  y un ataque puede llevar meses dentro

→ y por eso la retención de las copias más antiguas
  responde al escenario más lento, no al más rápido
→ típicamente: diarias 30 días, semanales 3 meses,
  mensuales 1 año, anuales lo que exija la norma
```

3. El ensayo de restauración

Es lo único que valida una copia, y es lo que casi nadie hace.

```
LO QUE NO VALIDA UNA COPIA
  que el trabajo de copia terminó con éxito
  que el fichero existe y tiene el tamaño esperado
  que el panel está en verde
  → todo eso puede ser cierto con una copia inservible
                                                ley 22

LO QUE SÍ
  restaurarla, ponerla a funcionar y comprobar que los
  datos sirven
```

Qué se comprueba en un ensayo:

```
1  QUE SE PUEDE RESTAURAR
   → y aquí aparecen los permisos que faltan, la clave que
     no está y el procedimiento que no existe

2  CUÁNTO TARDA, cronometrado
```

```
   → y ese número es el objetivo REAL, no el declarado
   → y suele ser varias veces mayor

3  CUÁNTO SE PIERDE, medido
   → la diferencia entre el momento de la copia y el del
     fallo

4  QUE LOS DATOS SIRVEN
   → no que el fichero se restauró: que la aplicación
     arranca, que las consultas devuelven lo esperado y
     que los recuentos cuadran
   → y esto exige una comprobación funcional, no técnica

5  Y QUE EL PROCEDIMIENTO LO PUEDE EJECUTAR OTRA PERSONA
   → quien lo escribió sabe los pasos que no escribió
                                        clases 128, 179
```

Y la disciplina:

```
ENSAYO PROGRAMADO, no cuando haya tiempo
  los conjuntos críticos, cada trimestre
  el resto, al menos una vez al año
  y siempre tras un cambio grande del sistema

EN UN ENTORNO SEPARADO
  → restaurar sobre producción para probar es como se
    convierte un ensayo en un incidente

Y CON EL RESULTADO PUBLICADO
  qué se restauró, cuánto tardó, qué falló
  → y los fallos se corrigen y se repite       clase 215
```

Y lo que los ensayos suelen encontrar:

```
la clave de cifrado no estaba disponible
el procedimiento tenía pasos que ya no existían
el plazo real era 3 o 4 veces el declarado
faltaba copiar algo: una tabla, un esquema, una
  configuración
la copia estaba, y era de un sistema que ya no existe así
y nadie tenía permiso para restaurar sin pedirlo

→ y ninguno de estos se detecta mirando el panel de copias
```

Y una forma barata de ensayar continuamente:

```
RESTAURACIÓN AUTOMÁTICA A UN ENTORNO DE PRUEBAS
  la copia de anoche se restaura sola en un entorno
  y se ejecutan unas comprobaciones funcionales
  → y si algo falla, alerta

→ cuesta poco y convierte el ensayo en algo continuo
→ y además da un entorno de pruebas con datos reales, que
  hay que tratar con sus permisos          clase 251
```

4. Lo que las copias no cubren

Hay escenarios que una copia no resuelve, y conviene saberlo antes.

```
EL BORRADO QUE SE DESCUBRE TARDE
  si la retención es de 30 días y el error se descubre a
  los 45, no hay de dónde volver
  → por eso la retención responde al escenario más lento

LA CORRUPCIÓN QUE SE PROPAGA A LAS COPIAS
  un error de aplicación que corrompe datos durante
  semanas
  → todas las copias de ese periodo tienen el error
  → y volver atrás pierde todo lo bueno de esas semanas
  → aquí lo que salva es detectarlo antes    clase 243

LO QUE NO SE PUEDE VOLVER A HACER
  correos enviados, pagos ejecutados, mensajes publicados
  → restaurar la base no los deshace       clase 246

LA PÉRDIDA DE LO QUE ESTABA EN VUELO
```

```
  las transacciones entre la última copia y el fallo
  → y ese es el objetivo de pérdida, que hay que aceptar
    explícitamente

Y EL SISTEMA QUE NO ESTÁ COPIADO
  → lo que no está en el inventario no está copiado
                                              clase 253
```

Y las prácticas que cubren esos huecos:

```
CONTRA EL DESCUBRIMIENTO TARDÍO
  detección de anomalías en los datos       clase 243
  y retención larga en las copias mensuales

CONTRA LA CORRUPCIÓN PROPAGADA
  comprobaciones de calidad que detienen    clase 243
  y capacidad de reprocesar desde el dato bruto
                                              clase 242

CONTRA LO IRREVERSIBLE
  límites y confirmación antes de actuar    clase 249

Y CONTRA LO NO COPIADO
  el inventario cruzado con la lista de copias
  → «recursos con datos que no tienen copia» es una
    consulta, y su resultado debe ser cero  clase 253
```

El plan de recuperación, con lo que debe contener:

```
por cada conjunto
  qué se copia, con qué frecuencia y retención
  dónde está y con qué credenciales se accede
  el objetivo de recuperación MEDIDO
  el procedimiento, probado por otra persona
  quién puede autorizar una restauración
  y en qué orden se restauran las cosas
    → una aplicación sin su base no sirve
    → y una base sin sus secretos, tampoco
```

Y lo que hay que vigilar:

```
copias con fallo, y su antigüedad
conjuntos con datos y SIN copia         → cero
antigüedad de la copia más reciente por conjunto  ley 13
fecha del último ensayo por conjunto
coste del almacenamiento de copias      clase 214
y cambios en la configuración de copias  → auditados
  → alguien que desactiva una programación es una señal
                                              clase 226
```

Y la lista de comprobación de la clase:

- se distingue réplica de copia, y no se confunden
- está escrito qué se copia por conjunto, con frecuencia y retención
- se copian estado del código, secretos, claves y artefactos
- las copias son inmutables durante un plazo
- viven en otra cuenta, con credenciales distintas
- producción no puede borrarlas ni cambiar su retención
- la comprobación de que no puede se ha ejecutado
- la clave de cifrado está disponible al restaurar
- hay retención larga para escenarios de descubrimiento tardío
- hay ensayo de restauración programado y publicado
- el ensayo comprueba que los datos SIRVEN
- lo ejecuta alguien que no escribió el procedimiento
- el objetivo de recuperación está medido, no declarado
- hay consulta de recursos con datos y sin copia
- los cambios de configuración de copias se auditan

Y el cierre que enlaza con la clase siguiente: con lo que existe inventariado, al día y recuperable, queda cómo se accede a ello para operarlo. Administración remota sin acceso permanente es la materia de la clase 256.

Ejemplo trabajado

CloudShop revisa sus copias. Lo que sigue es el ensayo que reveló que la clave de cifrado no estaba, los 41 conjuntos sin copia que nadie sabía, y el objetivo de recuperación que era cuatro veces el declarado.

El punto de partida:

```
el panel de copias                          todo en verde
trabajos de copia                                 118
  con éxito en los últimos 30 días                118

el objetivo declarado en el plan de continuidad
  plazo de recuperación                          1 hora
  pérdida máxima                                15 min

y ensayos de restauración ejecutados
  en los últimos 2 años                             1
  → sobre una base pequeña, en 2023
```

El primer ensayo completo.

```
se ensayó la restauración de la base de pedidos, la más
crítica

  09:00  se inicia
  09:04  se localiza la copia              ✓
  09:11  se solicita permiso de restauración
         → nadie del equipo lo tenía; hubo que escalar
  09:52  permiso concedido                 ✗ 41 min
  09:55  restauración iniciada
  10:38  restauración completada           ✓ 43 min
  10:40  la aplicación no arranca
         → los secretos de conexión estaban en el gestor de
           secretos de producción, que en el ensayo no era
           accesible
  11:20  resuelto copiando los secretos a mano
  11:25  la aplicación arranca
  11:30  COMPROBACIÓN FUNCIONAL
         → los recuentos no cuadran: faltaban 4 tablas
         → estaban en un esquema que la programación de
           copias no incluía

  ensayo detenido                           2 h 30
  resultado                                  FALLIDO

el objetivo declarado                       1 hora
el plazo real, con las correcciones          4 h 10
```

Y los cinco hallazgos:

```
1  nadie del equipo de guardia podía restaurar
   → 3 personas con permiso, por elevación temporal
                                                clase 230
2  los secretos no estaban en el plan de recuperación
   → añadidos, con su propia copia aislada
3  4 tablas fuera de la programación de copias
   → la programación se hizo por lista y el esquema nuevo
     no entró                                    ley 27
   → cambiada a «todo el esquema, con exclusiones
     declaradas»
4  el procedimiento tenía 3 pasos que ya no existían
5  y el plazo real era 4 veces el declarado
   → el plan de continuidad se corrigió con la cifra
     medida                                   clase 215
```

La clave de cifrado.

```
el segundo ensayo, esta vez simulando la pérdida de la
región principal

  la copia estaba replicada en la segunda región     ✓
  al intentar restaurar
    → la copia estaba cifrada con una clave gestionada por
      el cliente
```

```
    → y esa clave vivía SOLO en la región principal
                                                clase 239
    → la copia era ilegible

→ y esto no aparece en ningún panel: la copia existe, tiene
  el tamaño correcto y el trabajo terminó bien
                                                    ley 22

corrección
  la clave replicada a la segunda región
  y una comprobación mensual: restaurar un objeto pequeño
  desde la segunda región con la clave de allí
```

Los 41 conjuntos sin copia.

```
se cruzó el inventario con la lista de copias
                                                clase 253

  recursos con datos                                214
  con copia programada                              173
  SIN COPIA                                          41  ←

los 41
  14  almacenes de objetos creados por equipos, con datos
      de negocio
   9  bases de datos de entornos que se habían promovido a
      producción sin cambiar su configuración   ley 27
   6  conjuntos del almacén analítico con transformaciones
      que costaría semanas rehacer
   5  el ESTADO del código como infraestructura   clase 232
      → sin él, el código no sabe qué gestiona
   4  registros de decisión y documentación en una
      herramienta sin copias
   3  configuraciones de alertas y paneles       clase 238

→ y los 5 del estado eran el más grave: en un desastre, el
  código habría intentado recrear 4.100 recursos que ya
  existían                                    clase 232

corrección
  los 41 con copia
  y una consulta automática: «recursos con datos y sin
  copia» → debe dar cero
  → con alerta                                    ley 13
```

La protección de las copias.

```
la prueba negativa, desde credenciales de producción
  borrar una copia                  ✗ FUNCIONÓ
  cambiar su retención              ✗ FUNCIONÓ
  desactivar la programación        ✗ FUNCIONÓ

→ las copias vivían en la misma cuenta que los datos
→ y un ataque con credenciales de producción se las habría
  llevado                                   clase 166

corrección
  cuenta de copias separada, con credenciales propias
  producción escribe allí y NO puede leer, modificar ni
    borrar
  inmutabilidad de 35 días
  cambios de retención y de programación, con elevación
    temporal y alerta inmediata           clases 218, 226

la prueba, repetida
  borrar                            ✓ denegado
  cambiar retención                 ✓ denegado
  desactivar programación           ✓ denegado + alerta
```

La restauración continua, montada después.

```
cada noche
  la copia de las 4 bases críticas se restaura
  automáticamente en un entorno de pruebas
  se arranca la aplicación
```

```
  y se ejecutan 12 comprobaciones funcionales
    recuentos de tablas principales
    una consulta de negocio con resultado conocido
    y el arranque de los 3 servicios que dependen

  y si algo falla, alerta

en los primeros 6 meses
  fallos detectados                                  9
    4  una tabla nueva no estaba en la copia
    2  un cambio de esquema rompía la comprobación
    2  la copia de una noche estaba incompleta
    1  la clave rotó y el entorno de pruebas no la tenía

  → y los 9 se habrían descubierto en un desastre

y un efecto secundario
  el entorno de pruebas tiene datos reales de anoche
  → con datos personales, y con sus permisos y su
    seudonimización                          clase 251
  → se decidió seudonimizar en la restauración
```

El plan de recuperación, con el orden:

```
el primer ensayo completo del sistema descubrió que el
orden importaba

  orden correcto
    1  claves de cifrado
    2  secretos
    3  red y nombres                        clases 219, 231
    4  bases de datos
    5  almacenes de objetos
    6  artefactos e imágenes                clase 254
    7  estado del código como infraestructura clase 232
    8  aplicaciones
    9  y las comprobaciones funcionales

  → y en el primer intento, el equipo empezó por las
    aplicaciones
  → que no arrancaban porque no había base, ni secretos, ni
    imágenes
```

El resultado, al año:

```
conjuntos con datos y sin copia               41           0
ensayos de restauración al año                 1          14
fallos detectados por los ensayos              —           9
plazo de recuperación declarado            1 hora     2 h 40
plazo de recuperación MEDIDO            desconocido   2 h 40
pérdida medida                          desconocida    8 min
copias borrables desde producción             sí          no
clave disponible en la segunda región         no          sí
personas que pueden restaurar                  0           3
coste de almacenamiento de copias          1.400 €     2.100 €
```

Y la observación sobre la última fila:

```
el coste subió un 50 %
→ porque antes no se copiaban 41 conjuntos
→ y el plan de continuidad decía que sí
```

La lección que esta clase deja: el panel de copias estaba entero en verde con ciento dieciocho trabajos correctos, y la primera restauración real falló en cinco puntos distintos, incluido que nadie del equipo tenía permiso para restaurar. La copia de la segunda región existía, tenía el tamaño correcto y era ilegible, porque su clave de cifrado vivía solo en la región que se había perdido. Y el plazo declarado era de una hora: el medido, de cuatro.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-21-cloud-operations-automation/255-backups-restore-testing-vaults-e-inmutabilidad/
lab.py
```

El laboratorio selecciona el motor de práctica reliability 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 backup-restore 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 escenario de fallo con objetivo y recuperación medida. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye backup-restore 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 borrado deja los datos perdidos pese a tener replicación	La réplica protege de fallo de infraestructura, no de borrado: lo replica al instante	Ten copias de punto en el tiempo, además de réplicas, y distínguelas al declarar la protección.
Las copias se pueden borrar con las credenciales de producción	Viven en la misma cuenta y con los mismos permisos	Cuenta separada con credenciales propias, inmutabilidad por plazo y elevación temporal para cambiar retención o programación.
La copia existe y no se puede restaurar	La clave de cifrado vive solo en la región perdida	Replica la clave y comprueba mensualmente una restauración pequeña desde la otra región.
Un esquema o una tabla nueva no está en las copias	La programación se hizo por lista y lo nuevo no entra	Programa por conjunto completo con exclusiones declaradas, y cruza el inventario con la lista de copias.
El plazo de recuperación real es varias veces el declarado	El objetivo se declaró sin ensayar	Ensaya, cronometra los cinco tramos y corrige el plan con la cifra medida.
La restauración se completa y la aplicación no funciona	Se comprobó que el fichero se restauró, no que los datos sirven	Incluye comprobaciones funcionales en el ensayo y restaura en un entorno separado, con el orden de dependencias escrito.

■ 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. ¿De qué protege una réplica y de qué no?
2. ¿Qué se olvida copiar con más frecuencia?
3. ¿Cuáles son las tres protecciones de las copias y qué prueba las valida?
4. ¿Qué comprueba un ensayo de restauración además de que el fichero vuelve?
5. ¿Qué escenarios no cubren las copias y con qué se cubren?

Referencias

- AWS (2025). AWS Backup: vault lock and cross-account backups. <https://docs.aws.amazon.com/aws-backup/latest/devguide/vault-lock.html>
- Microsoft (2025). Azure Backup: immutable vaults and multi-user authorization. <https://learn.microsoft.com/en-us/azure/backup/backup-azure-immutable-vault-concept>
- Google Cloud (2025). Backup and DR Service. <https://cloud.google.com/backup-disaster-recovery/docs/concepts/overview>
- Beyer, B. y otros (2016). Site Reliability Engineering, cap. «Data integrity: what you read is what you wrote». <https://sre.google/sre-book/data-integrity/>
- NIST SP 800-34 (2010). Contingency planning guide for federal information systems. <https://csrc.nist.gov/pubs/sp/800/34/r1/upd1/final>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 21 en PDF · Manual integral

Anterior	Índice	Siguiente
← 254 · Patching, imágenes doradas y gestión de configuración	Parte 21 · Programa	256 · Administración remota sin SSH permanente →

Evaluación — 255 Backups, restore testing, vaults e inmutabilidad

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 backup-restore 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.

256 — Administración remota sin SSH permanente

Parte: 21 — Operación cloud, automatización y respuesta a incidentes Nivel: avanzado · Horas estimadas: 4
Laboratorio: security · Estado: EXECUTABLECORE

Propósito

Acceder a los sistemas para operarlos sin dejar puertas permanentes abiertas, que es el hueco que queda tras cerrar la red y montar la identidad. La clase explica por qué el acceso interactivo permanente es el patrón que hay que eliminar, cubre los mecanismos que lo sustituyen, y aborda lo que casi nadie resuelve: qué se hace cuando el mecanismo de acceso es precisamente lo que ha fallado.

Resultados de aprendizaje

Al finalizar podrás:

1. Eliminar el acceso interactivo permanente y las claves asociadas.
2. Elegir el mecanismo de acceso adecuado a cada necesidad.
3. Registrar y auditar lo que se hace durante un acceso.
4. Reducir la necesidad de entrar, que es la mejor solución.
5. Preparar el acceso de emergencia y probarlo.

Conceptos centrales

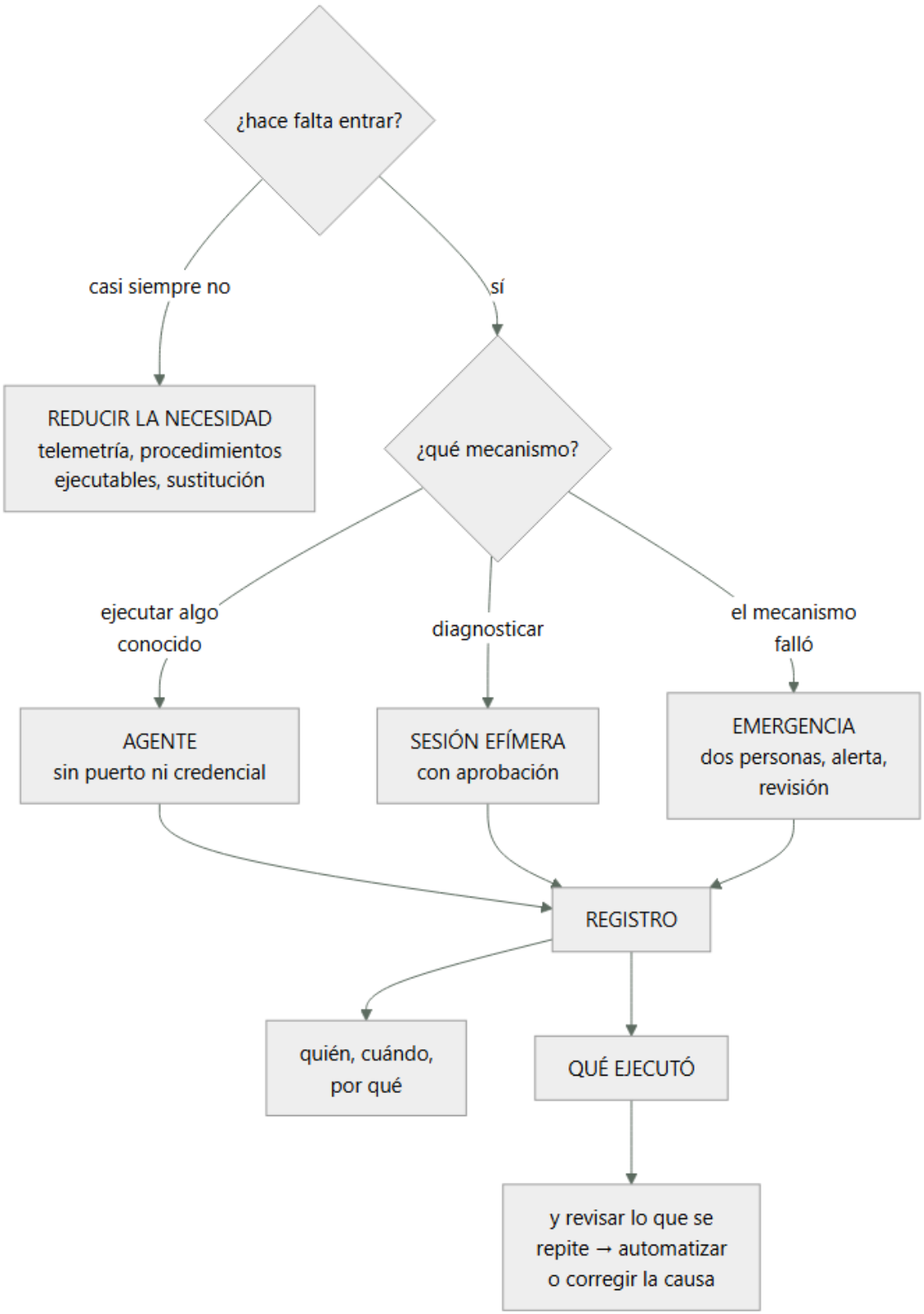
Concepto	Comprensión verificable
acceso interactivo	Sesión de línea de órdenes o de escritorio sobre un sistema. Lo que hay que hacer excepcional.
agente de administración	Componente del sistema que recibe órdenes de la plataforma, sin puerto abierto ni credencial.
sesión efímera	Acceso concedido por tiempo limitado, con justificación, aprobación y registro.
anfitrión intermedio	Punto único por el que pasa el acceso. Concentra el control y el registro, y es un punto único.
grabación de sesión	Registro de lo que se ejecuta durante un acceso. Necesario para auditar y para aprender.
acceso de emergencia	Vía alternativa cuando el mecanismo normal no funciona. Probada, vigilada y con dos personas.

Modelo mental

Operar es controlar cambios bajo incertidumbre: inventario, señal, diagnóstico, acción reversible y aprendizaje deben formar un ciclo cerrado.

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. Eliminar el acceso permanente

El patrón que hay que eliminar es concreto: un puerto de administración abierto y una credencial que vale siempre.

```
POR QUÉ ES UN PROBLEMA
  el puerto es un punto de entrada permanente
  la credencial no caduca y se comparte
```

```
  el acceso no queda justificado
  y lo que se hace dentro no queda registrado

→ y en la clase 226, el primer camino de ataque empezaba
  exactamente ahí: una máquina con el puerto de
  administración abierto
```

Los mecanismos que lo sustituyen, por orden de preferencia:

```
1  NO ENTRAR                                ← lo mejor
   la telemetría contesta la pregunta       clase 238
   el procedimiento se ejecuta desde la plataforma
                                                clase 259
   y el sistema se sustituye en vez de arreglarse
                                                clase 254

2  AGENTE DE ADMINISTRACIÓN
   el sistema tiene un agente que recibe órdenes de la
   plataforma
   + sin puerto abierto: la conexión la inicia el agente
   + sin credencial en el sistema: usa su identidad
   + con permisos por documento y con registro
   → y es lo que sustituye al 90 % de los accesos

3  SESIÓN EFÍMERA POR LA PLATAFORMA
   se abre una sesión interactiva a través del agente
   + sin puerto, con identidad de la nube y con registro
   + y con permiso concedido por tiempo limitado
                                                clase 218

4  ANFITRIÓN INTERMEDIO
   un punto por el que pasa todo el acceso
   – es un punto único, y hay que operarlo y protegerlo
   → sigue haciendo falta para sistemas que no admiten
     agente

5  ACCESO DIRECTO CON CLAVE
   → último recurso, con excepción registrada  ley 25
```

Y lo que hay que hacer con las claves de acceso:

```
las claves de sesión interactiva son credenciales de larga
duración
  → se comparten, se copian y no caducan
  → y su rotación es un procedimiento que nadie hace
                                                  ley 22

→ se eliminan, y el acceso pasa por identidad de la
  plataforma                              clases 206, 230
→ y las que queden, con certificado de corta duración
  emitido al pedir el acceso
```

Y la política que lo impone:

```
prohibir direcciones públicas en máquinas   clase 229
prohibir puertos de administración abiertos
  → y comprobarlo con una prueba negativa       ley 22
y obligar a que el agente esté instalado en la imagen base
                                                clase 254
```

2. Reducir la necesidad de entrar

La mejor solución al acceso no es controlarlo mejor: es no necesitarlo.

```
POR QUÉ SE ENTRA, en la práctica
  para MIRAR algo que no está en la telemetría
  para ejecutar un comando conocido
  para arreglar algo a mano
  y para investigar algo que no se entiende

→ y cada motivo tiene su solución
```

Y las soluciones, por motivo:

```
PARA MIRAR
  → lo que se mira repetidamente se convierte en una
```

```
    señal o en una consulta guardada       clases 238, 250
  → y si hace falta entrar para saber si algo funciona, la
    observabilidad tiene un hueco             clase 211

PARA EJECUTAR UN COMANDO CONOCIDO
  → procedimiento ejecutable desde la plataforma
                                                 clase 259
  → con parámetros validados y registro
  → y sin que nadie escriba comandos a mano

PARA ARREGLAR A MANO
  → casi siempre, lo correcto es sustituir  clase 254
  → y si el arreglo a mano se repite, es una causa que
    corregir, no una tarea que agilizar        clase 262

PARA INVESTIGAR
  → es el único motivo legítimo de una sesión interactiva
  → y aun así, con captura de lo que se ejecuta
```

Y la medida que dice si se está consiguiendo:

```
NÚMERO DE SESIONES INTERACTIVAS AL MES
  → y su tendencia
  → si no baja, no se está reduciendo la necesidad

y la segunda
  qué se ejecuta en esas sesiones
  → y lo que se repite es lo que hay que automatizar o
    corregir                                clase 259, 263
```

Y una advertencia:

```
SI ENTRAR ES MÁS DIFÍCIL QUE ÚTIL, SE BUSCARÁ OTRA VÍA
  una credencial guardada «por si acaso»
  un túnel abierto de forma permanente
  o una máquina de salto que nadie inventarió
                                          ley 16, clase 253
→ y por eso el acceso legítimo tiene que ser rápido
  → segundos cuando no requiere aprobación
```

3. Registrar y auditar

Un acceso sin registro deja al sistema sin respuesta ante la pregunta más simple: qué se hizo.

```
LO QUE HAY QUE REGISTRAR
  quién                    ← la persona, no la cuenta de
                             servicio        clase 230
  cuándo, y cuánto duró
  a qué sistema
  por qué: la justificación escrita
  quién lo aprobó, si hubo aprobación
  y QUÉ SE EJECUTÓ

→ y el último es el que casi nunca está
```

La grabación de sesión, con lo que aporta y lo que cuesta:

```
QUÉ APORTA
  saber qué se hizo, no solo que se entró
  investigar un incidente causado por un acceso
  y detectar lo repetitivo               clase 262

QUÉ CUESTA
  las sesiones contienen datos sensibles: consultas con
  datos personales, secretos escritos por error
  → se tratan como dato sensible: cifrado, acceso
    restringido, retención acotada       clases 238, 251
  y hay que decir a la gente que se graba

→ y grabar sin decirlo es un problema distinto
```

Y el destino, con la lección de la clase 238:

```
los registros de acceso van a un destino que el sistema
accedido NO puede modificar
  → en otra cuenta, con otras credenciales    clase 141
```

```
→ si no, quien entra puede borrar su rastro
```

Las alertas que corresponden:

```
uso del acceso de emergencia          → inmediata
acceso fuera de horario habitual
acceso a un sistema crítico
sesión de duración anómala
acceso desde una ubicación inhabitual
y comandos de una lista sensible ejecutados en sesión
  → borrados masivos, cambios de permisos, desactivación
    de registros                              clase 226
```

Y la revisión, que es lo que convierte el registro en mejora:

```
MENSUAL
  ¿cuántas sesiones hubo y por qué?
  ¿qué se ejecutó, y qué se repite?
  → y lo que se repite pasa a ser un procedimiento
    ejecutable                                clase 259
  ¿alguien accede a sistemas que ya no le corresponden?
  → y eso se cruza con la revisión de accesos clase 230
```

Y una comprobación que da información inesperada:

```
¿A QUÉ SISTEMAS NO HA ENTRADO NADIE EN 6 MESES?
  → o están perfectamente automatizados
  → o nadie los necesita y son candidatos a retirar
                                        clases 253, 254
```

4. Cuando el mecanismo es lo que falla

Es el escenario que casi nadie prepara: el acceso normal no funciona precisamente cuando hace falta.

```
CUÁNDO OCURRE
  el proveedor de identidad no responde   clase 218
  el agente de administración no arranca
  la red que da acceso está caída          clase 219
  la política condicional bloquea a todos  clase 218
  o el propio sistema de acceso está en la región perdida
                                                clase 215

→ y en todos esos casos, el incidente que hay que arreglar
  incluye no poder entrar a arreglarlo
```

El acceso de emergencia, con lo que lo hace real:

```
DOS VÍAS, no una
  porque una puede estar afectada por el mismo fallo

CON CREDENCIALES GUARDADAS FÍSICAMENTE
  largas, en un sitio con acceso controlado
  y con dos personas necesarias para usarlas, si el sistema
  es crítico

EXCLUIDAS de las políticas condicionales   clase 218
  → y esa exclusión, COMPROBADA

CON ALERTA INMEDIATA en cada uso
  → a varias personas, por varios canales

Y CON REVISIÓN POSTERIOR OBLIGATORIA
  qué se hizo, por qué hizo falta y qué se cambia para que
  no vuelva a hacer falta
```

Y la prueba, que es lo que este programa ha visto fallar:

```
USAR EL ACCESO DE EMERGENCIA, CADA TRIMESTRE
  y cronometrar

  en la clase 179, estaba creado, documentado y NO
  funcionaba
  en la clase 218, la política condicional lo había
  incluido sin querer
  en la clase 230, había perdido un permiso al reorganizar
  la jerarquía                                  ley 27
```

```
→ tres veces, en tres nubes distintas
→ y las tres se descubrieron probándolo         ley 22
```

Y lo que hay que tener escrito antes:

```
quién puede usarlo y en qué circunstancias
dónde están las credenciales y quién las custodia
qué hacer si la persona que las custodia no está
cómo se avisa
y qué se hace después: rotar las credenciales usadas
```

Y una decisión de diseño que reduce la necesidad:

```
QUE EL SISTEMA DE ACCESO NO DEPENDA DE LO QUE PUEDE FALLAR
  no alojarlo en la misma región que protege
  no depender del mismo proveedor de identidad para la
    emergencia
  y no depender de la red que puede estar caída

→ y esto es la aritmética de dependencias de la clase 185,
  aplicada al propio acceso
```

Y la lista de comprobación de la clase:

- no hay puertos de administración abiertos, comprobado
- no hay direcciones públicas en máquinas sin justificar
- no hay claves de sesión de larga duración
- el agente de administración está en la imagen base
- el acceso interactivo requiere permiso temporal
- hay anfitrión intermedio solo donde el agente no llega
- el acceso legítimo tarda segundos cuando no requiere aprobación
- se registra quién, cuándo, por qué y QUÉ EJECUTÓ
- el registro va a un destino que el sistema no puede tocar
- las grabaciones se tratan como dato sensible y se avisa
- hay alerta de emergencia, horario anómalo y comandos sensibles
- se revisa mensualmente qué se ejecuta y qué se repite
- hay dos vías de acceso de emergencia
- están excluidas de las políticas, comprobado
- se prueban cada trimestre y se cronometran
- el sistema de acceso no depende de lo que puede fallar
- se mide el número de sesiones interactivas y su tendencia

Y el cierre que enlaza con la clase siguiente: con el acceso resuelto, queda quién se entera cuando algo va mal y qué hace. Alertas, guardia, escalado y comunicación es la materia de la clase 257.

Ejemplo trabajado

CloudShop elimina el acceso permanente a sus sistemas. Lo que sigue son los 118 puertos abiertos que nadie había inventariado, la revisión mensual que convirtió 41 sesiones en 3, y el acceso de emergencia que no funcionó por tercera vez.

El punto de partida:

```
máquinas con puerto de administración accesible      118
  desde internet                                      14  ←
  desde la red corporativa                           104

claves de sesión interactiva                         210
  compartidas entre varias personas                   88
  con más de 2 años                                  140
  cuya persona ya no está en la empresa               31  ←

sesiones interactivas al mes                         410
registradas                                            0
con justificación                                      0
```

Y las 14 desde internet:

```
  9  máquinas de salto de distintos equipos
     → cada equipo montó la suya                clase 253
  3  servidores heredados que «necesitaban acceso directo»
```

```
  1  una instancia de pruebas de 2023             ley 25
  1  la que aparecía como primer paso del camino de ataque
     de la clase 226
```

La migración, en cuatro fases.

```
FASE 1 · el agente en la imagen base
  el agente de administración pasa a la imagen base
                                                clase 254
  → y con la renovación programada, en 4 semanas estaba en
    el 96 % de las máquinas
  → el 4 % restante: 11 máquinas que no se reconstruyen
    → agente instalado a mano, con su fecha de revisión

FASE 2 · el acceso por la plataforma
  sesiones a través del agente, con identidad de la nube
  permiso concedido por elevación temporal   clase 218
    sin aprobación: activación en 15 segundos
    a sistemas críticos: aprobación de 2, en minutos
  y registro de la sesión completa

FASE 3 · cerrar los puertos
  se midió primero: qué puertos recibían conexiones y de
  dónde
    conexiones en 30 días                          1.940
    desde las nuevas sesiones                      1.610
    desde direcciones conocidas de personas          280
    DESDE DIRECCIONES DESCONOCIDAS                    50  ←
      → 41 de un proveedor de monitorización que nadie
        recordaba                                 ley 20
      →  9 de un rastreador automatizado externo

  y entonces se cerraron
    puertos abiertos                          118 → 0
    y las 3 máquinas heredadas, tras un anfitrión
    intermedio con registro

FASE 4 · eliminar las claves
  210 → 6
    los 6, de sistemas que no admiten otra cosa
    con certificado de corta duración, emitido al pedir
    acceso
    y excepción registrada con fecha            ley 25

  y las 31 de personas que ya no estaban: revocadas
```

Y la prueba negativa:

```
✓  conectar al puerto de administración desde internet
                                                denegado
✓  conectar desde la red corporativa            denegado
✓  usar una clave revocada                      denegado
✓  abrir sesión sin permiso temporal            denegado
✓  abrir sesión a un sistema crítico sin aprobación
                                                denegado
```

La revisión mensual, y lo que cambió.

```
mes 1, tras montar el registro
  sesiones                                        341
  y lo que se ejecutaba, agrupado

    reiniciar un servicio                          88
    mirar registros de una aplicación              71
    consultar el espacio en disco                  54
    limpiar ficheros temporales                    41
    reprocesar mensajes fallidos                   29
                                                clase 210
    ampliar una cuota                              22
    investigar algo no rutinario                   19
    y otros                                        17

→ 19 de 341 eran investigación
→ el resto era trabajo repetitivo         clase 262
```

Y lo que se hizo con cada grupo:

```
MIRAR REGISTROS (71)
  → los registros ya estaban centralizados; la gente
    entraba por costumbre                       clase 238
  → formación y consultas guardadas: 71 → 4

CONSULTAR ESPACIO EN DISCO (54)
  → no había señal de disco en el panel         clase 211
  → añadida, con alerta: 54 → 0

LIMPIAR TEMPORALES (41)
  → la causa era una rotación de registros mal configurada
  → CORREGIDA LA CAUSA: 41 → 0

REINICIAR UN SERVICIO (88)
  → procedimiento ejecutable desde la plataforma
                                                clase 259
  → y al analizarlo: 61 de los 88 eran el mismo servicio
    con una fuga de memoria
  → corregida la fuga: 88 → 6

REPROCESAR MENSAJES (29)
  → procedimiento ejecutable, con límite de ritmo
                                                clase 210
  → 29 → 0 sesiones (se ejecuta desde la plataforma)

AMPLIAR CUOTA (22)
  → petición automatizada                     clase 262
  → 22 → 0

INVESTIGACIÓN (19)
  → se mantienen: es el motivo legítimo
```

Y el resultado:

```
sesiones interactivas                          341          14
  de investigación                              19          11
  de trabajo repetitivo                        322           3
```

Y la observación del equipo:

```
de las 322 sesiones repetitivas
  se automatizaron                                 117
  se ELIMINARON corrigiendo la causa               163
  quedaron                                           3

→ más de la mitad no hacía falta automatizarlas: hacía
  falta arreglar lo que las provocaba
→ y el instinto del equipo había sido automatizarlas todas
                                                clase 262
```

El acceso de emergencia, que falló otra vez.

```
prueba trimestral, primera ejecución tras la migración

  09:00  se simula que el proveedor de identidad no
         responde
  09:02  se recuperan las credenciales de emergencia del
         sobre físico                                ✓
  09:05  se intenta entrar
         → la cuenta existía y estaba excluida de las
           políticas condicionales                   ✓
         → pero el permiso para abrir sesión por el agente
           dependía de un grupo del directorio
         → y el directorio era lo que no respondía

  → la vía de emergencia dependía de lo que había fallado
                                                clase 185

  09:20  se usa la segunda vía: el anfitrión intermedio con
         clave local
         → funcionó                                  ✓
         → tiempo total                          20 min
```

```
correcciones
  la primera vía se rehace con permisos locales, no
  dependientes del directorio
  y la segunda se mantiene, con su clave rotada tras cada
  uso

y la observación
  es la TERCERA vez que el acceso de emergencia falla en
  este programa: clases 179, 218 y ahora
  → y las tres se descubrieron probándolo         ley 22
  → nunca por un incidente real, afortunadamente
```

El registro y su tratamiento:

```
cada sesión graba
  quién, cuándo, cuánto, a qué, por qué y qué ejecutó

y la grabación
  cifrada, en la cuenta de seguridad          clase 238
  producción no puede leerla ni borrarla
  retención de 180 días
  acceso restringido a 3 personas, auditado
  y avisado a todo el equipo, por escrito

en los 6 meses siguientes
  consultas a las grabaciones                        14
    en investigación de incidentes                   11
    en auditoría                                      2
    y 1 por una reclamación de un cliente
      → «¿quién modificó este pedido a mano?»
      → contestado en 4 minutos
```

El resultado, al año:

```
puertos de administración abiertos            118           0
  desde internet                               14           0
claves de sesión de larga duración            210           6
  de personas que ya no están                  31           0
sesiones interactivas al mes                  410          14
  registradas                                   0        100 %
  con justificación                             0        100 %
sesiones de trabajo repetitivo                322           3
conexiones desde direcciones desconocidas      50           0
tiempo para conceder acceso legítimo      variable       15 s
acceso de emergencia probado                    no    trimestral
  vías independientes                            1           2
```

La lección que esta clase deja: cerrar ciento dieciocho puertos fue lo fácil; lo que redujo el riesgo de verdad fue pasar de trescientas cuarenta y una sesiones al mes a catorce, y de esas trescientas veintidós que sobraban, más de la mitad se eliminaron corrigiendo la causa, no automatizándolas. Y el acceso de emergencia falló por tercera vez en este programa, otra vez por una dependencia no contada: dependía del directorio, que era justamente lo que había fallado.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-21-cloud-operations-automation/256-administracion-remota-sin-ssh-permanente/lab.py
```

El laboratorio selecciona el motor de práctica security 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 secure-operations 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 control con amenaza, mitigación y evidencia. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye secure-operations 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
Hay puertos de administración abiertos que nadie recuerda	Cada equipo montó su propia máquina de salto y nadie las inventarió	Inventaría, mide qué conexiones reciben y de dónde, y sustituye por agente con sesiones efímeras antes de cerrar.
Quedan claves de acceso de personas que ya no están	Son credenciales de larga duración compartidas y sin rotación	Elimina las claves, usa identidad de la plataforma y, donde no se pueda, certificados de corta duración emitidos al pedir acceso.
El equipo busca vías alternativas para entrar	El acceso legítimo tarda o requiere aprobación siempre	Activación en segundos cuando no requiere aprobación, y aprobadores con turnos cuando sí.
No se sabe qué se hizo durante un acceso	Se registra que se entró, no lo que se ejecutó	Graba la sesión, guárdala fuera del alcance del sistema accedido y trátala como dato sensible, avisando al equipo.
Se automatizan tareas que no deberían existir	Se agiliza el síntoma en vez de corregir la causa	Revisa mensualmente qué se ejecuta y qué se repite; corrige la causa antes de automatizar.
El acceso de emergencia no funciona cuando hace falta	Depende del mismo componente que ha fallado	Dos vías independientes, excluidas de las políticas, con permisos locales, probadas y cronometradas cada trimestre.

■ 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 patrón de acceso que hay que eliminar y por qué?
2. ¿Cuáles son los cinco mecanismos, por orden de preferencia?
3. ¿Qué se hace con cada uno de los cuatro motivos por los que se entra?
4. ¿Qué hay que registrar además de quién entró y dónde debe guardarse?

5. ¿Qué hace real un acceso de emergencia y con qué frecuencia se prueba?

Referencias

- AWS (2025). Systems Manager Session Manager. <https://docs.aws.amazon.com/systems-manager/latest/userguide/session-manager.html>
- Microsoft (2025). Azure Bastion and just-in-time VM access. <https://learn.microsoft.com/en-us/azure/bastion/bastion-overview>
- Google Cloud (2025). Identity-Aware Proxy for TCP forwarding. <https://cloud.google.com/iap/docs/using-tcp-forwarding>
- NIST SP 800-207 (2020). Zero Trust Architecture. <https://csrc.nist.gov/pubs/sp/800/207/final>
- Beyer, B. y otros (2018). The Site Reliability Workbook, cap. sobre eliminación de trabajo repetitivo. <https://sre.google/workbook/eliminating-toil/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 21 en PDF · Manual integral

Anterior	Índice	Siguiente
← 255 · Backups, restore testing, vaults e inmutabilidad	Parte 21 · Programa	257 · Alertas, on-call, escalamiento y comunicación →

Evaluación — 256 Administración remota sin SSH permanente

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 secure-operations 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.

257 — Alertas, on-call, escalamiento y comunicación

Parte: 21 — Operación cloud, automatización y respuesta a incidentes Nivel: avanzado · Horas estimadas: 4
Laboratorio: incident · Estado: EXECUTABLECORE

Propósito

Organizar quién se entera cuando algo va mal, qué hace y a quién avisa. La clase aborda las alertas desde la única pregunta que las justifica —¿qué hace quien la recibe?—, monta la guardia de forma sostenible, fija el escalado y desarrolla la parte que este programa ha visto dominar el tiempo de los incidentes: comunicar, que es trabajo y hay que repartirlo.

Resultados de aprendizaje

Al finalizar podrás:

1. Diseñar alertas accionables y retirar las que no lo son.
2. Montar una guardia sostenible, con carga medida.
3. Escalar con criterios claros y sin depender de que alguien esté.
4. Comunicar durante un incidente, con papeles repartidos.
5. Medir la salud de la guardia y actuar sobre ella.

Conceptos centrales

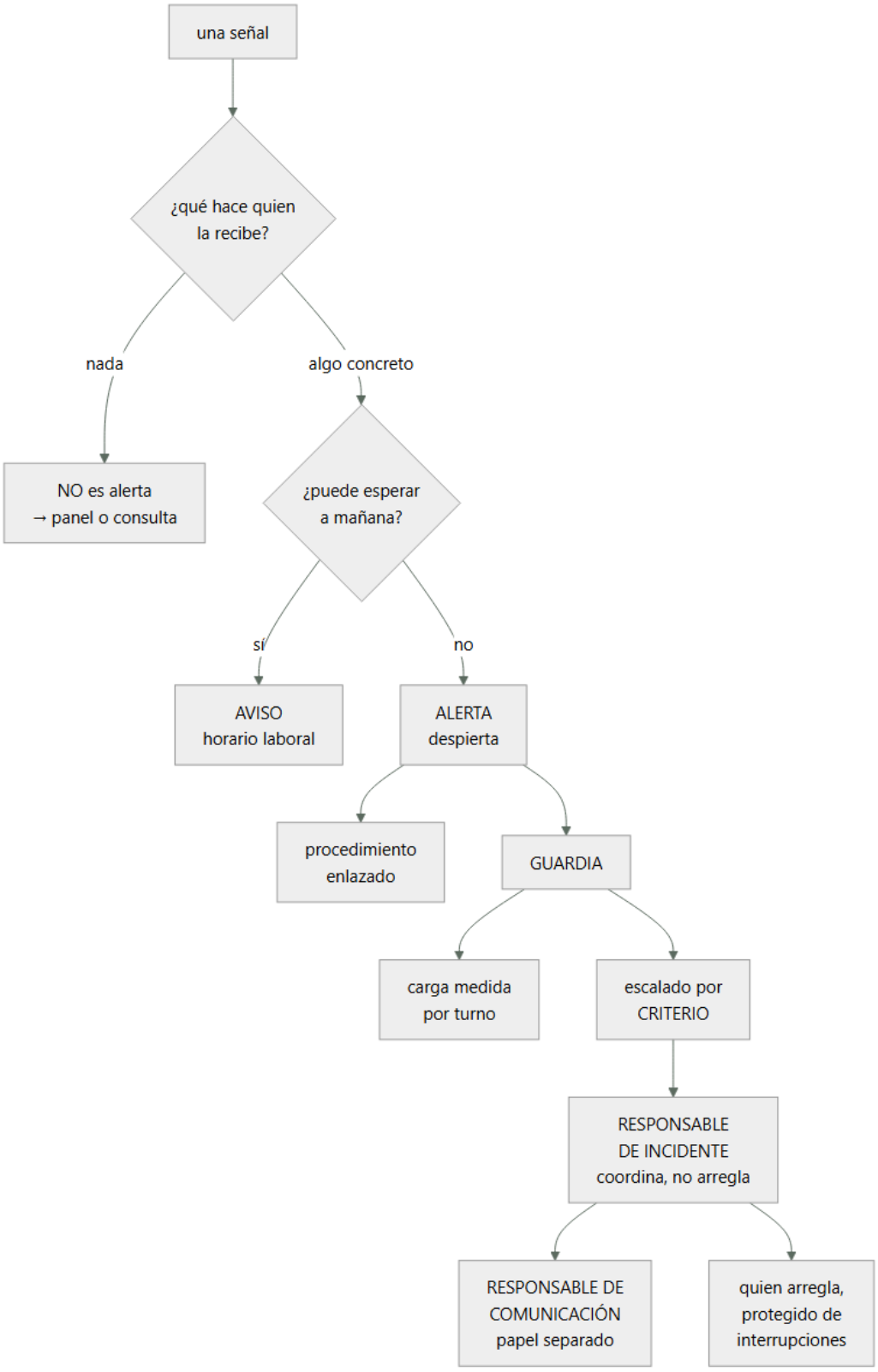
Concepto	Comprensión verificable
alerta accionable	La que exige una acción concreta de quien la recibe. Si no la exige, no es una alerta.
carga de guardia	Interrupciones por turno, dentro y fuera de horario. Determina si la guardia es sostenible.
escalado	Paso del incidente a alguien con más contexto o más autoridad, por criterio y no por desesperación.
responsable de incidente	Quien coordina. No arregla: decide, reparte y protege al que arregla.
responsable de comunicación	Quien informa a los afectados y a la organización. Papel separado, porque es trabajo a tiempo completo.
fatiga de alertas	Estado en que las alertas dejan de producir acción porque hay demasiadas.

Modelo mental

Operar es controlar cambios bajo incertidumbre: inventario, señal, diagnóstico, acción reversible y aprendizaje deben formar un ciclo cerrado.

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. La única pregunta que justifica una alerta

Toda la disciplina de alertas cabe en una pregunta, y casi ninguna organización la hace.

```
¿QUÉ HACE QUIEN LA RECIBE?

  si la respuesta es «mirar y ver que no pasa nada»
```

```
    → no es una alerta: es ruido
  si es «esperar a ver si se arregla sola»
    → no es una alerta
  si es «avisar a otro»
    → la alerta debe ir a ese otro
  si es «ejecutar este procedimiento»
    → es una alerta, y el procedimiento va enlazado
  y si es «investigar porque los usuarios están sufriendo»
    → es una alerta
```

Y la segunda pregunta, que decide si despierta:

```
¿PUEDE ESPERAR A MAÑANA?
  sí  → aviso, en horario laboral
  no  → alerta, y despierta

→ y la mayoría de lo que hoy despierta puede esperar
→ despertar a alguien tiene un coste real: al día
  siguiente rinde menos y el turno siguiente empieza peor
```

Sobre qué se alerta, con la regla de las clases 123 y 211:

```
SE ALERTA SOBRE SÍNTOMAS DE USUARIO
  el objetivo de nivel de servicio consume presupuesto
  demasiado rápido                          clase 238
  el flujo crítico no funciona
  el retraso de proceso supera lo acordado

Y SOBRE AUSENCIA Y ANTIGÜEDAD
  lo que dejó de ocurrir                        ley 13
  lo que lleva demasiado tiempo sin actualizarse

NO SE ALERTA SOBRE
  causas intermedias: CPU, memoria, disco
    → salvo que la acción sea inmediata y concreta
  errores sueltos
  y cosas que se recuperan solas
```

Y el error que produce la fatiga:

```
SE ALERTA SOBRE CUANTO SE PUEDE MEDIR
  porque medir es fácil y decidir es difícil
  → cientos de alertas
  → y entonces se ignoran, incluidas las que importan
                                              clase 125

→ y el síntoma es una cifra: la proporción de alertas que
  resultan ser algo
  → por debajo del 50 %, el sistema ya no funciona
```

Y la limpieza, que hay que hacer periódicamente:

```
cada alerta, una vez al trimestre
  ¿se ha disparado?
    no, en un año  → o sobra o su umbral está mal
  ¿fue accionable?
  ¿tiene procedimiento enlazado?
  ¿llegó a alguien que hizo algo?

→ y en las clases 211, 225 y 238, más de la mitad se
  retiraron sin perder nada
```

2. La guardia, sostenible

Una guardia insostenible se degrada sola: la gente se va, y quien queda hace peor su trabajo.

```
LO QUE HACE FALTA MEDIR
  interrupciones por turno, dentro y fuera de horario
  cuántas despiertan
  duración de cada intervención
  y cuántos turnos seguidos hace cada persona

→ y un umbral que este programa sostiene
  más de 2 interrupciones por turno fuera de horario
  significa que hay un problema, no que hace falta más
  gente                                     clase 125
```

El tamaño del turno, con la aritmética:

```
una rotación necesita al menos 6 personas para que nadie
esté de guardia más de una semana de cada seis
  → con 4, cada dos semanas de cada cuatro: se quema
  → con menos, no es una guardia: es que hay una persona
    disponible siempre                              ley 20

y si el equipo no llega
  se comparte la guardia con otro equipo
  o se reduce la superficie: menos servicios, menos alertas
                                                  clase 253
  → y no se finge que hay guardia
```

Y las condiciones que la hacen sostenible:

```
COMPENSACIÓN, en tiempo o en dinero, acordada
DERECHO A DESCONECTAR tras una noche mala
SIN OTRAS TAREAS durante el turno
  → estar de guardia y tener entregas es tener dos trabajos
Y UN LÍMITE: si el turno se pasa arreglando cosas, el
  trabajo siguiente es reducir las alertas, no aguantar
```

Lo que hay que tener antes de que empiece un turno:

```
el procedimiento de cada alerta, enlazado    clase 125
los accesos concedidos y PROBADOS         clase 256
  → y no descubrir a las 3 que no se puede entrar
el camino esperado de los flujos             clase 194
las consultas de diagnóstico guardadas       clase 238
la lista de a quién escalar y cómo
y el traspaso del turno anterior
```

Y el traspaso, que es barato y se olvida:

```
al terminar el turno, un resumen escrito
  qué pasó
  qué quedó a medias
  qué está degradado
  y qué esperar

→ sin él, quien entra empieza a ciegas
→ y los problemas que se arrastran no se ven
```

Y una práctica que reduce mucho la carga:

```
EL TURNO REVISA SUS PROPIAS ALERTAS
  al terminar, quien estuvo de guardia dice qué alertas
  fueron ruido
  → y esas se corrigen o se retiran esa semana
  → y quien sufre el ruido es quien decide sobre él
                                                  clase 172
```

3. Escalar y coordinar

El escalado falla cuando depende de que alguien esté disponible y de que a quien está se le ocurra.

```
ESCALAR POR CRITERIO, no por desesperación
  «si en 15 minutos no hay hipótesis, se escala»
  «si afecta a más de N usuarios, se escala»
  «si toca datos o dinero, se escala inmediatamente»
  «si hay que decidir algo que no me corresponde, se
   escala»

→ y el criterio se escribe antes
→ así escalar no es admitir una derrota: es seguir el
  procedimiento
```

Y la cadena, con lo que debe garantizar:

```
nivel 1   quien está de guardia
nivel 2   quien conoce ese servicio
nivel 3   quien puede decidir sobre el negocio

y cada nivel
  con más de una persona
  con un canal que funcione aunque el sistema esté caído
```

```
  y con un plazo: si no responde en N minutos, se pasa al
    siguiente automáticamente

→ y ese plazo automático es lo que evita el «llamé y no
  contestó»
```

Los papeles durante un incidente, que es lo que más ordena:

```
RESPONSABLE DE INCIDENTE
  coordina; NO arregla
  decide qué se prueba y qué no
  reparte el trabajo
  y protege a quien arregla de las interrupciones
  → y este papel es el que más se salta, porque parece que
    no hace nada

QUIEN ARREGLA
  una o dos personas, sin interrupciones
  → y si son cinco, se estorban

RESPONSABLE DE COMUNICACIÓN
  informa a los afectados y a la organización
  → papel SEPARADO, porque comunicar es trabajo a tiempo
    completo
  → y si lo hace quien arregla, deja de arreglar

Y QUIEN TOMA NOTAS
  la línea de tiempo, mientras ocurre
  → reconstruirla después cuesta el triple  clase 127
```

Y la regla que se aprende cara:

```
EN UN INCIDENTE GRANDE, EL RESPONSABLE NO TOCA EL TECLADO
  en cuanto empieza a arreglar, deja de coordinar
  → y nadie coordina

→ y en incidentes pequeños, una persona hace los tres
  papeles; el criterio para separarlos es el tamaño
```

Y el canal, con su requisito:

```
un canal por incidente, no una conversación general
  → todo el mundo ve lo mismo y queda registrado
  y un canal ALTERNATIVO que no dependa del sistema
    afectado
  → si el incidente afecta a la identidad, la herramienta
    de mensajería puede no funcionar         clase 218
  → y eso hay que haberlo probado                ley 22
```

4. Comunicar

Este programa ha medido varias veces que decidir y comunicar dominan el tiempo total de un incidente. Comunicar es trabajo.

```
A QUIÉN HAY QUE INFORMAR
  los usuarios afectados
  atención al cliente, que recibe las llamadas
  el negocio, que tiene que decidir cosas
  los socios, si les afecta
  y la organización, para que no pregunten uno a uno

→ y cada uno necesita algo distinto
```

Y lo que cada uno necesita:

```
USUARIOS
  qué no funciona, en sus términos
  qué pueden hacer mientras tanto
  y cuándo habrá noticias
  → NO les interesa la causa técnica

ATENCIÓN AL CLIENTE
  lo anterior, más qué responder a las preguntas concretas
  → y lo antes posible, porque están recibiendo llamadas
    ya
```

```
NEGOCIO
  el impacto: cuántos, cuánto, desde cuándo
  y qué decisiones hay que tomar

LA ORGANIZACIÓN
  que hay un incidente, quién lo lleva y dónde mirar
  → y así dejan de preguntar en canales aleatorios
```

El ritmo, que es lo que más tranquiliza:

```
la primera comunicación, en los primeros minutos
  aunque no se sepa nada: «lo sabemos, estamos en ello»
actualizaciones a intervalo FIJO y anunciado
  «volveremos a informar en 30 minutos»
  → y se cumple, aunque no haya novedades
  → «sin novedades» es información

→ y lo que genera desconfianza no es el incidente: es el
  silencio
```

Y lo que no hay que hacer:

```
prometer un plazo de resolución que no se sabe
  → «estará en 20 minutos» y a las dos horas sigue
  → mejor: «la siguiente actualización en 30 minutos»
minimizar el impacto
y explicar la causa técnica antes de estar seguro
  → y luego corregirla, que es peor
```

Y las plantillas, que ahorran minutos que no se tienen:

```
mensajes preparados para
  detección, actualización, mitigación y cierre
  por cada audiencia
→ y se rellenan, no se redactan
```

Después del incidente, con lo que enlaza con la clase 261:

```
el cierre se comunica igual que el inicio
y la revisión se hace con la línea de tiempo tomada
durante                                    clase 127

y las preguntas de la revisión
  ¿cuánto tardamos en enterarnos?
  ¿cuánto en decidir?
  ¿cuánto en comunicar?
  y ¿qué señal habría avisado antes?
```

Y lo que hay que vigilar de la guardia:

```
interrupciones por turno, y cuántas despiertan
proporción de alertas que resultan ser algo
tiempo hasta la primera acción
tiempo hasta la primera comunicación
escalados y si respetaron el criterio
y rotación: cuántas personas, y si alguien repite de más
```

Y la lista de comprobación de la clase:

- ■ cada alerta tiene respuesta a «¿qué hace quien la recibe?»
- ■ lo que puede esperar es aviso, no alerta
- ■ se alerta sobre síntomas, ausencia y antigüedad
- ■ cada alerta tiene procedimiento enlazado
- ■ se mide la proporción que resulta ser algo
- ■ se revisan y retiran alertas cada trimestre
- ■ la rotación tiene personas suficientes
- ■ se mide la carga por turno
- ■ quien está de guardia no tiene otras entregas
- ■ los accesos están probados antes del turno
- ■ hay traspaso escrito entre turnos
- ■ el turno revisa y corrige sus propias alertas
- ■ el escalado tiene criterio escrito y plazo automático
- ■ hay papeles separados de coordinación, arreglo y comunicación
- ■ hay canal alternativo probado
- ■ hay plantillas por audiencia y ritmo fijo de

```
  actualización
```

Y el cierre que enlaza con la clase siguiente: con la guardia montada y la comunicación repartida, queda lo que hace quien arregla: diagnosticar. El método de triaje por capas es la materia de la clase 258.

Ejemplo trabajado

CloudShop reorganiza su guardia. Lo que sigue son las 890 alertas mensuales de las que 8 % eran algo, la persona que hacía guardia una semana de cada dos, y el incidente en que comunicar consumió más tiempo que arreglar.

El punto de partida:

```
alertas configuradas                              412
disparadas al mes                                 890
que resultaron ser algo                        71 (8 %)

rotación de guardia
  personas                                          4
  → una semana de cada dos para cada una
interrupciones por turno
  media                                          27,4
  fuera de horario                                9,1
  que despertaron                                 4,2

bajas y salidas del equipo en 12 meses              2
  → las dos citaron la guardia en la entrevista de
    salida
```

La limpieza de alertas.

```
las 412, revisadas con la pregunta «¿qué hace quien la
recibe?»

  nada, o mirar y comprobar que va bien           184
    → retiradas
  esperar a ver si se arregla sola                 61
    → retiradas; y 14 se convirtieron en aviso con umbral
      más alto
  avisar a otro equipo                             38
    → redirigidas a ese equipo
  ejecutar un procedimiento                        72
    → mantenidas, con procedimiento enlazado
  investigar porque hay impacto                    57
    → mantenidas

alertas tras la limpieza                          167
  de ellas, que despiertan                         41
  el resto, avisos en horario laboral              126
```

Y la segunda pasada, con la pregunta del horario:

```
de las 41 que despertaban
  ¿puede esperar a mañana?
    sí                                             23
      → pasaron a aviso
    no                                             18

alertas que despiertan                        41 → 18
```

Y el resultado del primer mes:

```
alertas configuradas                          412         167
disparadas al mes                             890         104
que resultaron ser algo                    71 (8 %)    88 (85 %)
interrupciones por turno                     27,4         3,1
  fuera de horario                            9,1         1,2
  que despertaron                             4,2         0,4
```

Y la observación:

```
las alertas que resultaron ser algo SUBIERON de 71 a 88
→ porque las 184 de ruido estaban tapando cosas reales
→ y el equipo había dejado de mirar               ley 15
```

La rotación.

```
con 4 personas, una semana de cada dos
  → insostenible, y demostrado con las dos salidas

se plantearon tres opciones
  contratar                                6 meses
  compartir con otro equipo                posible
  reducir la superficie                    posible

lo que se hizo, las tres cosas
  1  compartir la guardia con el equipo de datos
     → de 4 a 9 personas: una semana de cada nueve
     → y con formación cruzada: 6 sesiones
  2  reducir la superficie
     → 22 servicios de los 61 se retiraron o pasaron a
       gestionados                  clases 253, 254
     → y las alertas de esos, con ellos
  3  y la contratación siguió su curso, sin urgencia

semanas de guardia por persona y año
  antes                                      26
  después                                   5,8
```

Y las condiciones que se acordaron:

```
compensación en tiempo: un día por semana de guardia
derecho a desconectar la mañana siguiente a una noche con
  más de una interrupción
sin entregas comprometidas durante el turno
y la regla que más cambió las cosas
  «si un turno tiene más de 5 interrupciones fuera de
   horario, el trabajo de la semana siguiente es reducir
   alertas»
  → aplicada 3 veces en el primer trimestre, 0 en el
    segundo
```

El traspaso, que no existía.

```
se añadió un resumen escrito al terminar el turno
  qué pasó
  qué quedó a medias
  qué está degradado
  y qué esperar

en los primeros 3 meses
  problemas que se arrastraban entre turnos y nadie veía
                                                      7
    · una réplica con retraso creciente desde hacía 9 días
    · un trabajo que fallaba cada noche y alguien
      relanzaba
    · y 5 más

→ y los 7 llevaban semanas: cada turno los arreglaba y no
  se los contaba al siguiente                  ley 15
```

El incidente en que comunicar dominó.

```
incidente   el flujo de pago dejó de funcionar para el 40 %
            de los usuarios

  14:02  alerta de presupuesto de error         clase 238
  14:04  primera acción: se revisa el panel
  14:09  hipótesis: un despliegue de las 13:58
  14:12  reversión iniciada
  14:19  servicio restablecido

  → 17 minutos de detección a resolución

y lo que pasó en paralelo
  14:06  atención al cliente empieza a recibir llamadas
  14:11  el director de operaciones pregunta en un canal
  14:14  el equipo comercial pregunta en otro
  14:16  un socio escribe
  14:22  dirección pregunta por el impacto

  y quien estaba revirtiendo el despliegue contestaba a
```

```
  todos
  → la reversión se retrasó 6 minutos por eso

  y después
  14:19-15:40  reconstruir qué había pasado, calcular el
               impacto y contestar a cada uno
               → 1 h 21 de comunicación tras 17 minutos de
                 incidente

→ y ese reparto es el que este programa ha medido varias
  veces                                    clases 179, 215
```

Y lo que se montó después:

```
PAPELES SEPARADOS, a partir de cierta gravedad
  responsable de incidente: coordina, no toca el teclado
  quien arregla: 1 o 2, sin interrupciones
  responsable de comunicación: informa a todos
  y quien toma notas

  → y para incidentes menores, una persona hace los cuatro

CANAL POR INCIDENTE
  se abre uno; todas las preguntas van ahí
  → y las preguntas en otros canales se redirigen

PLANTILLAS por audiencia
  usuarios, atención al cliente, negocio, socios y
  organización
  → 4 momentos cada una: detección, actualización,
    mitigación y cierre

RITMO FIJO
  primera comunicación en menos de 10 minutos
  actualizaciones cada 30, aunque no haya novedades

y la línea de tiempo, tomada DURANTE
```

Y el efecto, medido en los 9 incidentes siguientes:

```
tiempo hasta la primera comunicación         n/d       7 min
tiempo de quien arregla dedicado a
  comunicar                                  38 %        3 %
tiempo total de comunicación tras el
  incidente                                1 h 21      18 min
canales distintos con preguntas                5           1
reconstrucción de la línea de tiempo       2-3 h        0
  (se toma durante)
```

El canal alternativo, probado.

```
se simuló un incidente que afectaba al proveedor de
identidad
  → la herramienta de mensajería del equipo usa ese mismo
    proveedor                                  clase 218
  → nadie podía entrar

  el canal alternativo: un grupo en un servicio de
  mensajería independiente, con los teléfonos
  → primera prueba: 4 de 9 personas no lo tenían
    instalado
  → corregido, y probado cada trimestre          ley 22
```

El escalado, con criterio.

```
antes   se escalaba cuando quien estaba de guardia se
        rendía
        y llamaba a quien creía que sabía
        → y a veces no contestaba

después
  criterios escritos
    sin hipótesis en 15 min                → nivel 2
    afecta a más de 1.000 usuarios         → nivel 2 y
                                             comunicación
```

```
    toca datos o dinero                    → inmediato
    hay que decidir algo de negocio        → nivel 3
  cadena con 2 personas por nivel
  y plazo automático: sin respuesta en 5 min, siguiente

y en 6 meses
  escalados                                          14
  que respetaron el criterio                         14
  llamadas sin respuesta que bloquearon               0
```

El resultado, al año:

```
alertas configuradas                          412         151
disparadas al mes                             890          94
proporción accionable                         8 %         89 %
interrupciones por turno                     27,4         2,3
que despertaron                               4,2         0,3
semanas de guardia por persona y año           26         5,8
salidas del equipo citando la guardia            2           0
tiempo hasta la primera comunicación         n/d         7 min
problemas arrastrados entre turnos              7           0
```

La lección que esta clase deja: retirar ciento ochenta y cuatro alertas que no exigían ninguna acción hizo que las que resultaban ser algo subieran de setenta y una a ochenta y ocho: el ruido estaba tapando problemas reales. Y en el incidente que mejor se resolvió técnicamente —diecisiete minutos de detección a reversión—, comunicar consumió una hora y veintiún minutos después, y retrasó seis minutos la propia reversión porque quien arreglaba contestaba a cinco canales a la vez.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-21-cloud-operations-automation/257-alertas-on-call-escalamiento-y-comunicacion/
lab.py
```

El laboratorio selecciona el motor de práctica incident 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 oncall-system 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 cronología, roles, comunicación y aprendizaje. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye oncall-system 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
Hay cientos de alertas y nadie las mira	Se alerta sobre todo lo que se puede medir, sin preguntar qué acción exige	Pregunta qué hace quien la recibe; lo que no exige acción concreta es un panel, no una alerta.
La gente abandona el equipo por la guardia	Rotación con pocas personas y carga alta por turno	Comparte guardia, reduce superficie y mide interrupciones por turno; por encima del umbral, el trabajo es reducir alertas.
Problemas que se arrastran semanas sin que nadie los vea	No hay traspaso entre turnos y cada uno los apaga sin contarlo	Resumen escrito al terminar el turno, con lo que quedó a medias y lo degradado.
Quien arregla el incidente pierde tiempo contestando preguntas	No hay papeles separados ni canal único	Separa coordinación, arreglo y comunicación a partir de cierta gravedad, y abre un canal por incidente.
Se escala tarde y a quien no contesta	El escalado depende de la desesperación y de la disponibilidad	Criterios escritos, dos personas por nivel y paso automático al siguiente si no responden en el plazo.
Durante un incidente nadie puede comunicarse	El canal de mensajería depende del sistema afectado	Ten un canal alternativo independiente y pruébalo cada trimestre.

■ 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 la única pregunta que justifica una alerta?
2. ¿Qué umbral de interrupciones por turno indica que hay un problema?
3. ¿Por qué el responsable de incidente no debe tocar el teclado?
4. ¿Qué necesita cada audiencia durante un incidente?
5. ¿Qué genera desconfianza más que el propio incidente?

Referencias

- Beyer, B. y otros (2016). Site Reliability Engineering, cap. «Being on-call». <https://sre.google/sre-book/being-on-call/>
- Google (2018). The Site Reliability Workbook: alerting on SLOs. <https://sre.google/workbook/alerting-on-slos/>
- PagerDuty (2025). Incident response documentation: roles and escalation. <https://response.pagerduty.com/>
- Atlassian (2025). Incident communication best practices. <https://www.atlassian.com/incident-management/incident-communication>
- Allspaw, J. (2016). Incident command for IT. <https://www.oreilly.com/library/view/incident-management-for/9781492045922/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 21 en PDF · Manual integral

Evaluación — 257 Alertas, on-call, escalamiento y comunicación

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 oncall-system 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.

258 — Triage de red, cómputo, datos y dependencias

Parte: 21 — Operación cloud, automatización y respuesta a incidentes Nivel: avanzado · Horas estimadas: 4
Laboratorio: incident · Estado: EXECUTABLECORE

Propósito

Diagnosticar bajo presión con un método, en vez de con intuición. La clase da el orden de descarte que resuelve la mayoría de los casos en minutos, las preguntas que discriminan por capa —red, cómputo, datos y dependencias—, y las trampas de razonamiento que alargan los incidentes: quedarse con la primera hipótesis, confundir correlación con causa y no comprobar lo que se da por sabido.

Resultados de aprendizaje

Al finalizar podrás:

1. Aplicar un orden de descarte que no dependa de la intuición.
2. Discriminar por capa con preguntas que separan causas.
3. Usar la línea de cambios como primera hipótesis, siempre.
4. Evitar las trampas de razonamiento que alargan los incidentes.
5. Decidir cuándo mitigar sin haber entendido la causa.

Conceptos centrales

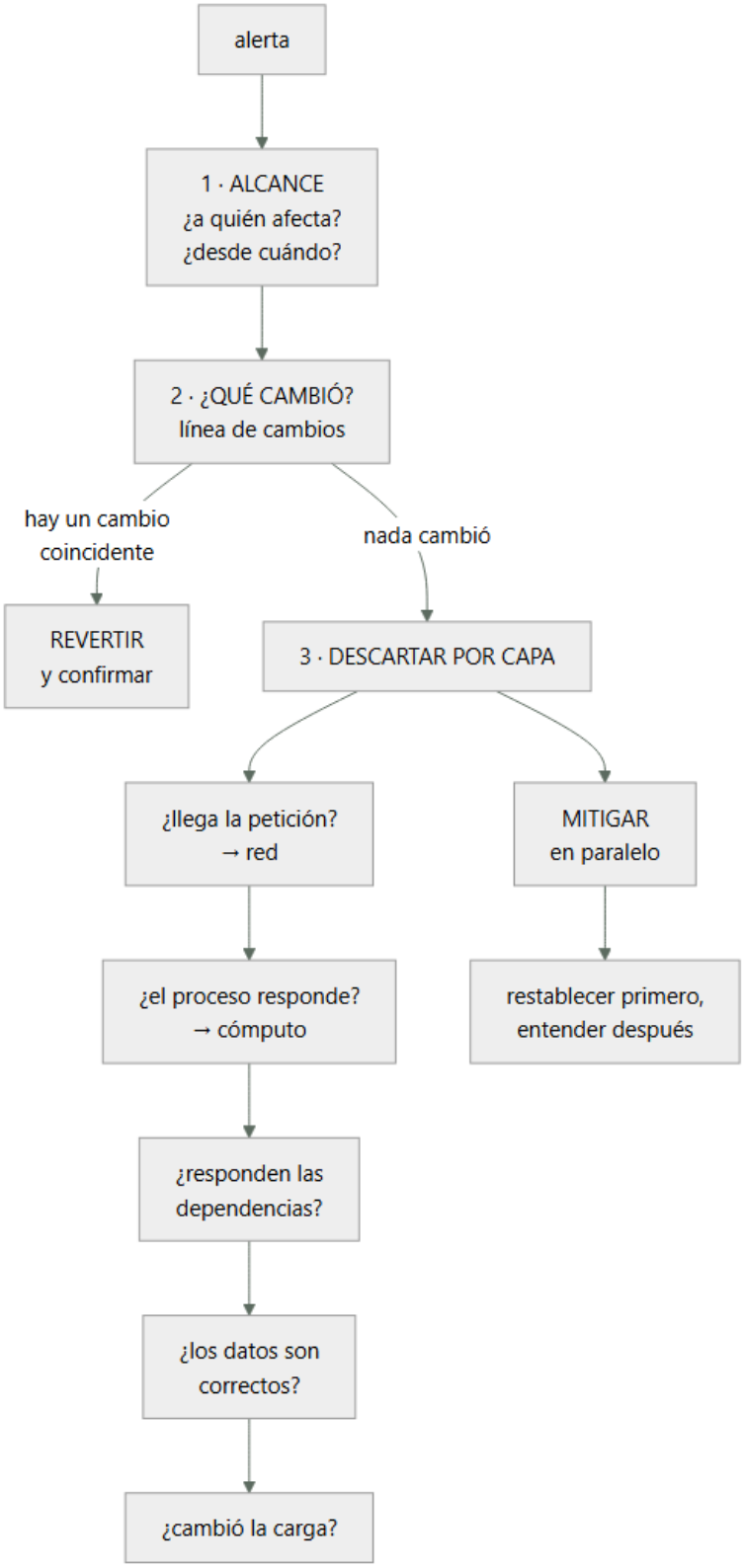
Concepto	Comprensión verificable
triaje	Clasificar rápido el alcance y la capa afectada antes de investigar en profundidad.
línea de cambios	Registro de qué se ha desplegado, configurado o modificado, con hora. La primera hipótesis.
descarte	Eliminar capas con una comprobación barata, en vez de buscar la causa directamente.
mitigar	Restablecer el servicio sin haber entendido la causa. Casi siempre lo correcto primero.
sesgo de la primera hipótesis	Quedarse con la primera explicación plausible y buscar solo pruebas que la apoyen.
correlación temporal	Que dos cosas ocurran a la vez. Es una pista, no una causa.

Modelo mental

Operar es controlar cambios bajo incertidumbre: inventario, señal, diagnóstico, acción reversible y aprendizaje deben formar un ciclo cerrado.

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 orden de descarte

Bajo presión, la intuición lleva a mirar donde uno sabe mirar, no donde está el problema. El orden lo evita.

```
1  ALCANCE
   ¿a quién afecta? ¿a todos, a una región, a un tipo de
   usuario, a un cliente?
```

```
   ¿desde cuándo, exactamente?
   ¿es total o parcial?

   → y el alcance ya descarta capas enteras
     todos, de golpe        → algo global: cambio,
                              dependencia común, red
     una región             → esa región
     un porcentaje estable  → una parte de las réplicas
     un cliente             → sus datos o su configuración

2  ¿QUÉ CAMBIÓ?
   la línea de cambios, en la ventana anterior
   → despliegues, configuración, banderas, políticas,
     rutas, certificados, cuotas
   → y también cambios de OTROS: proveedores, socios,
     equipos vecinos

   → esta es la primera hipótesis SIEMPRE
   → y en la mayoría de los incidentes, la última también

3  DESCARTAR POR CAPA, con comprobaciones baratas
   → y en el orden en que el tráfico las atraviesa

4  Y EN PARALELO, MITIGAR
   → no se espera a entender para restablecer
```

Y por qué la línea de cambios va antes que cualquier otra cosa:

```
la mayoría de los incidentes los causa un cambio
  un despliegue                              clase 102
  una configuración                          clase 232
  un certificado que caduca                  clase 196
  una política nueva                         clase 217
  una ruta                                   clase 194
  una cuota que se alcanza                   clase 217
  o un cambio de un tercero                  clase 188

→ y si hay un cambio coincidente, revertirlo es más rápido
  que entender
→ y si al revertir se arregla, la hipótesis queda
  confirmada
```

Y lo que hace posible ese paso:

```
UNA LÍNEA DE CAMBIOS ÚNICA, de todas las fuentes
  despliegues, infraestructura, banderas, políticas,
  cambios manuales
  → y de todos los equipos

→ y si cada equipo tiene la suya, este paso no se puede
  dar                                        clase 122
→ es de las cosas más baratas de montar y de las que más
  tiempo ahorran
```

Y cuándo mitigar sin entender:

```
CASI SIEMPRE
  revertir, desviar tráfico, reiniciar, escalar capacidad,
  desactivar una función
  → el servicio vuelve y la investigación sigue en calma

CUÁNDO NO
  si mitigar destruye la evidencia que hace falta
  → entonces se captura primero: registros, volcados,
    estado                                   clase 202
  y si mitigar puede empeorar: un reinicio que pierde datos
    en memoria

→ y la decisión de mitigar la toma quien coordina, no quien
  investiga                                  clase 257
```

2. Discriminar por capa

Cada capa tiene una pregunta que la descarta con poco esfuerzo. El orden es el del recorrido del tráfico.

```
RED                                          clase 202
  ¿llega la petición al destino?
    → registros de flujo: ¿aceptada o rechazada?
  ¿qué ruta se aplica?
    → comprobación de siguiente salto
  ¿resuelve el nombre, y a qué?
    → y desde LA RED que importa            clase 195
  ¿es simétrico el camino?

  firma característica
    funciona en un sentido        asimetría + cortafuegos
    funciona a ratos              dos caminos, uno roto
    todo a la vez y de golpe      sesión BGP o cuota
    lo pequeño sí y lo grande no  MTU               clase 198

CÓMPUTO
  ¿el proceso está vivo y responde?
  ¿cuántas instancias sanas hay frente a las esperadas?
  ¿se reinició? ¿por qué?
    → memoria, comprobación de salud, expulsión
                                          clases 213, 221
  ¿está estrangulado por cuota o por límite?
    → y aquí la CPU del nodo engaña          clase 213

  firma característica
    latencia alta con recursos ociosos   límite o
                                         estrangulamiento
    reinicios silenciosos                memoria
    p99 malo y p50 bien                  una réplica
                                         degradada

DEPENDENCIAS                                 clase 185
  ¿responden? ¿y con qué latencia?
  ¿alguna responde LENTO en vez de fallar?
    → el fallo gris, que la redundancia no cubre
  ¿se agotó algún grupo de conexiones o de puertos?
                                        clases 207, 221
  ¿hay reintentos multiplicando la carga?  clase 201

DATOS                                        clase 243
  ¿los datos son correctos, o solo están?
  ¿cambió la distribución?                      ley 29
  ¿hay un flujo parado?
  ¿un despliegue cambió un esquema?

CARGA
  ¿cambió el tráfico? ¿en volumen o en FORMA?
  ¿hay un cliente nuevo, un rastreador, un bucle de
    reintentos?
  ¿estamos cerca del codo?                    clase 186
```

Y la regla que ordena el uso de herramientas:

```
de lo barato a lo caro                       clase 202
  paneles y señales agregadas
  registros de flujo y consultas guardadas
  trazas del periodo
  registros de las trazas
  y solo entonces, captura o perfilado

→ y empezar por lo caro es el error más común bajo presión
```

3. Las trampas de razonamiento

La mayoría de los incidentes largos no lo son por falta de datos: lo son por errores de razonamiento.

```
1  QUEDARSE CON LA PRIMERA HIPÓTESIS
   se encuentra una explicación plausible y se busca solo
   lo que la confirma
   → y si es falsa, se pierden horas

   la corrección
     escribir DOS hipótesis alternativas antes de investigar
     y decir qué observación distinguiría entre ellas
```

```
     → «si es A, veremos X; si es B, veremos Y»

2  CONFUNDIR CORRELACIÓN CON CAUSA
   «empezó justo cuando desplegamos» es una pista fuerte
   → y a veces el despliegue solo hizo visible algo que ya
     estaba

   la corrección
     revertir y comprobar
     → si al revertir se arregla, la relación es causal
     → si no, la hipótesis era falsa y hay que soltarla

3  NO COMPROBAR LO QUE SE DA POR SABIDO
   «eso no puede ser, lo cambiamos la semana pasada»
   «esa alerta funciona»
   «ese servicio no lo usa nadie»
   → y este programa ha demostrado las tres falsas
                              clases 179, 211, 253

   la corrección
     comprobar lo barato, aunque parezca obvio
     → 30 segundos de comprobación frente a una hora de
       suposición

4  BUSCAR DONDE UNO SABE MIRAR
   quien sabe de red mira la red; quien sabe de bases,
   la base
   → y el orden de descarte lo evita

5  ASUMIR UNA SOLA CAUSA
   los incidentes grandes suelen tener dos o tres cosas a
   la vez
   → una que degrada y otra que impide detectarlo
   → y arreglar una y ver que sigue mal desconcierta

6  Y NO PARAR A PENSAR
   tras 20 minutos sin avanzar, seguir probando cosas es
   peor que parar 3 minutos y reordenar
   → y quien coordina es quien debe forzar esa pausa
                                              clase 257
```

Y una técnica que funciona muy bien y cuesta nada:

```
DECIR EN VOZ ALTA LO QUE SE ESTÁ HACIENDO
  «voy a comprobar si la petición llega al destino»
  «esperaba ver X y veo Y»

→ obliga a explicitar la hipótesis
→ permite que otro detecte el error
→ y produce la línea de tiempo sin esfuerzo   clase 257
```

Y la pregunta que desatasca más incidentes:

```
«¿QUÉ ESTAMOS DANDO POR SUPUESTO?»
  → y comprobar lo primero que salga

y la segunda
«¿QUÉ OBSERVACIÓN NOS HARÍA CAMBIAR DE IDEA?»
  → si no hay ninguna, la hipótesis no es útil
```

4. Los casos difíciles

Algunos patrones no ceden al orden normal y conviene reconocerlos.

```
INTERMITENTE
  falla el 3 % de las veces, sin patrón aparente
  → casi siempre es UNA de N: una réplica, una zona, un
    nodo, un camino                    clases 196, 202
  → y la forma de encontrarlo es agrupar por dimensión
    ¿todas las peticiones fallidas van a la misma
    instancia? ¿a la misma zona? ¿al mismo destino?

LENTO EN VEZ DE ROTO
  el fallo gris                                clase 185
  → nada da error; los percentiles altos se degradan
```

```
  → y la redundancia no lo cubre: el destino sigue «sano»
  → lo detecta la expulsión de atípicos y el reparto por
    conexiones en vuelo                       clase 196

CÍCLICO
  ocurre a la misma hora, o cada N horas
  → un trabajo programado, una rotación, una caducidad,
    una renegociación                         clase 198
  → y la pregunta es «¿qué ocurre con esa periodicidad?»

EMPEORA CON EL TIEMPO
  una fuga: memoria, conexiones, ficheros, hilos
  → y el síntoma es que reiniciar lo arregla temporalmente
  → si reiniciar arregla, es una fuga hasta que se
    demuestre lo contrario

SOLO EN PRODUCCIÓN
  volumen, datos reales, concurrencia, o una diferencia de
  configuración                                clase 104
  → y la primera comprobación es la diferencia de
    configuración entre entornos

Y SOLO PARA UN CLIENTE
  sus datos, su configuración, su volumen o su clave de
  partición                              clases 208, 223
```

Cuando no se encuentra, que también ocurre:

```
si el servicio está restablecido y no se encuentra la
causa
  se documenta lo que se descartó
  se añade la señal que habría ayudado    clase 238
  y se cierra, con la investigación abierta

→ y eso es mejor que mantener a todo el equipo horas
  buscando con el servicio ya funcionando
→ y peor que no hacerlo: el problema volverá
```

Y lo que hay que dejar preparado para el siguiente:

```
la consulta que se acabó usando, GUARDADA  clase 238
la señal que faltaba, añadida
el procedimiento actualizado con lo aprendido
y el caso, añadido a las pruebas negativas si procede
                                          clases 216, 250

→ y así el triaje siguiente es más corto
```

Y la lista de comprobación de la clase:

- hay línea de cambios única, de todas las fuentes
- el triaje empieza por alcance y por «¿qué cambió?»
- el descarte sigue el recorrido del tráfico
- se usa la herramienta barata antes que la cara
- se mitiga en paralelo, sin esperar a entender
- se captura evidencia antes de mitigar si hace falta
- se escriben dos hipótesis y qué las distinguiría
- se comprueba lo que se da por sabido
- se dice en voz alta lo que se está haciendo
- se para a reordenar si no se avanza en 20 minutos
- los intermitentes se agrupan por dimensión
- si reiniciar arregla, se investiga como fuga
- cuando no se encuentra, se documenta lo descartado
- la consulta usada se guarda y la señal que faltaba se añade

Y el cierre que enlaza con la clase siguiente: cuando el diagnóstico y la respuesta se repiten, escribirlos no basta: hay que poder ejecutarlos. Procedimientos ejecutables y remediación automática es la materia de la clase 259.

Ejemplo trabajado

Cuatro incidentes de CloudShop, resueltos con el método. Lo que sigue es el que se resolvió en cuatro minutos por la línea de cambios, el que costó tres horas por quedarse con la primera hipótesis, y el intermitente que resultó ser una de doce.

Incidente 1 · Cuatro minutos, por la línea de cambios.

```
03:14  alerta: presupuesto de error del flujo de compra
       consumiéndose 14 veces más rápido

03:14  ALCANCE
       todos los usuarios, de golpe, desde las 03:02
       → algo global

03:15  ¿QUÉ CAMBIÓ?
       línea de cambios, ventana 02:45-03:05
         02:58  despliegue del servicio de precios
         03:01  cambio de una regla del cortafuegos
         03:03  rotación programada de un certificado

03:16  tres candidatos; el cambio del cortafuegos es el más
       coincidente con las 03:02

03:17  se revierte el cambio del cortafuegos
03:18  restablecido

       tiempo total                              4 min

y la confirmación
  la regla nueva denegaba el tráfico hacia el servicio de
  precios desde una subred creada la semana anterior
                                       clases 219, 231
```

Y la observación:

```
sin la línea de cambios unificada, el equipo habría
empezado por el despliegue de precios, que también
coincidía
→ y habría revertido lo que no era
→ la hora exacta del cambio fue lo que discriminó
                                                clase 122
```

Incidente 2 · Tres horas, por la primera hipótesis.

```
11:20  el listado de catálogo con p99 de 4.100 ms
       el p50, normal

11:22  primera hipótesis: «la base está lenta»
       → alguien había visto la CPU de la base al 71 %

11:22-13:40  se investiga la base
  se revisan consultas lentas          nada anómalo
  se añade una réplica de lectura      no mejora
  se revisan índices                   correctos
  se sube el tamaño de la instancia    no mejora

13:45  alguien pregunta: «¿todas las peticiones lentas van
       a la misma instancia?»
13:47  se agrupa por instancia
       → el 100 % de las peticiones lentas iba a 1 de las
         12 instancias del servicio
13:50  esa instancia tenía un disco degradado
13:52  retirada; p99 a 240 ms

       tiempo total                           2 h 32
       tiempo desde la pregunta correcta       7 min
```

Y el análisis posterior:

```
la CPU de la base al 71 % era su valor NORMAL en esa hora
  → correlación tomada por causa
  → y nadie comprobó si era anómala             clase 122
```

```
y la pregunta que faltó al principio
  «¿el fallo es de todas las instancias o de una?»
  → que es la pregunta del intermitente y del p99 malo con
    p50 bueno                                clase 196

correcciones
  el panel de servicio muestra el p99 POR INSTANCIA
  y se añadió expulsión de atípicos          clase 196
  → esa combinación habría resuelto el caso sola
```

Incidente 3 · El intermitente.

```
síntoma   entre el 2 % y el 4 % de las llamadas al servicio
          de inventario fallaban por plazo
          sin patrón horario, desde hacía 9 días

el método
  se agruparon las peticiones fallidas por cada dimensión
  disponible

    por instancia de origen        repartido
    por instancia de destino       repartido
    por zona de origen             repartido
    por zona de DESTINO            41 % en una zona
    por nodo                       repartido
    por partición de la base       repartido
    por CLIENTE                    repartido

  → una zona concentraba el 41 % de los fallos con el 33 %
    del tráfico

  y dentro de esa zona
    por instancia de destino       1 de 4 concentraba el
                                   94 % de los fallos de
                                   la zona

  → una de doce instancias, en una zona
  → su comprobación de salud pasaba porque respondía al
    punto de vida                          clase 196
  → y su punto de listo no comprobaba la base

tiempo de diagnóstico con el método        22 min
tiempo que llevaba abierto                  9 días
```

Y la lección:

```
los intermitentes casi siempre son UNA DE N
→ y agrupar por cada dimensión disponible es mecánico
→ y se puede dejar como consulta guardada    clase 238
```

Incidente 4 · Dos causas a la vez.

```
16:40  el procesamiento de pedidos se detiene

16:42  ALCANCE: total, desde las 16:31
16:43  ¿QUÉ CAMBIÓ? nada en la ventana
16:45  DESCARTE POR CAPA
  ¿llega la petición?          sí
  ¿el proceso responde?        sí
  ¿dependencias?               la cola no entrega
                                          clase 237
16:50  la cola tenía 41.000 mensajes sin confirmar
       y el consumidor estaba parado
16:52  se reinicia el consumidor
16:54  empieza a consumir; se restablece

17:10  y vuelve a pararse

17:12  segunda hipótesis: hay algo que lo para
  → un mensaje con un campo que el consumidor no maneja
  → y ese mensaje bloqueaba la clave de ordenación
                                          clase 237
17:20  el mensaje se aparta; el consumo se normaliza

y la SEGUNDA causa
```

```
  ¿por qué nadie se enteró a las 16:31?
  → la alerta de antigüedad de la cola existía
  → con umbral de 24 horas                          ley 15
  → y llevaba 9 minutos cuando el negocio lo notó

→ dos cosas: una que rompió y otra que impidió detectarlo
→ arreglar la primera y ver que volvía a pasar fue lo que
  llevó a la segunda
```

Lo que quedó montado tras los cuatro:

```
LÍNEA DE CAMBIOS ÚNICA
  despliegues, infraestructura, banderas, políticas, rutas,
  certificados y cambios manuales     clases 122, 232, 256
  → en un solo sitio, con hora

CONSULTAS GUARDADAS de triaje
  «agrupar peticiones fallidas por cada dimensión»
  «cambios en la última hora»
  «instancias sanas frente a esperadas»
  «dependencias y su latencia»
  «¿hay algún flujo parado?»
  → enlazadas desde los procedimientos      clase 257

PANELES POR INSTANCIA, no solo agregados
  → el p99 agregado ocultaba una de doce

Y LA PLANTILLA DE TRIAJE
  alcance · qué cambió · descarte por capa · dos hipótesis
  · qué las distingue
  → escrita, y usada en cada incidente
```

El efecto, en los 6 meses siguientes:

```
tiempo medio hasta la hipótesis correcta   52 min       8 min
incidentes resueltos revirtiendo un cambio    n/d      11/19
tiempo medio hasta mitigar                 1 h 20      14 min
incidentes con más de una causa               n/d        4/19
  detectados los dos                          n/d        4/4
intermitentes abiertos más de 2 días            3          0
incidentes en que se empezó por captura         6          0
  de paquetes
```

Y una cifra que el equipo destacó:

```
de 19 incidentes, 11 se resolvieron revirtiendo un cambio
en menos de 15 minutos
→ el 58 %
→ y todos ellos habrían sido investigaciones largas sin la
  línea de cambios
```

La lección que esta clase deja: el incidente que se resolvió en cuatro minutos y el que costó dos horas y media tenían la misma dificultad técnica: la diferencia fue preguntar «¿qué cambió?» y «¿es una de N?» antes de investigar. Y en el que llevaba nueve días abierto, agrupar por cada dimensión disponible —una tarea mecánica de veintidós minutos— encontró una instancia de doce cuya comprobación de salud pasaba porque solo miraba si el proceso estaba vivo.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-21-cloud-operations-automation/258-triage-de-red-computo-datos-y-dependencias/lab.
py
```

El laboratorio selecciona el motor de práctica incident 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 diagnostic-tree 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 cronología, roles, comunicación y aprendizaje. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye diagnostic-tree 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 investiga durante horas y la causa era un cambio reciente	No se consultó la línea de cambios al principio, o cada equipo tiene la suya	Monta una línea de cambios única de todas las fuentes y consúltala como primera hipótesis siempre.
Se persigue una hipótesis falsa mucho tiempo	Se tomó la primera explicación plausible y se buscó solo lo que la confirmaba	Escribe dos hipótesis y qué observación las distingue; y si revertir no arregla, suelta la hipótesis.
Un fallo intermitente lleva días sin diagnosticar	No se agrupó por dimensiones para ver si es una de N	Agrupa las peticiones fallidas por instancia, zona, destino, partición y cliente; casi siempre concentra en una.
El p99 es pésimo con el p50 bien y no se encuentra la causa	Los paneles son agregados y ocultan la instancia degradada	Mide por instancia y añade expulsión de atípicos; una réplica lenta no falla la comprobación de salud.
Se arregla la causa y el problema vuelve	Había dos causas: una que rompía y otra que impedía detectarlo	Tras mitigar, pregunta siempre por qué no se detectó antes; suele haber una segunda causa.
Se empieza capturando paquetes o volcados	Bajo presión se recurre a la herramienta que uno domina	Sigue el orden de descarte y usa las herramientas de lo barato a lo caro.

■ 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. ¿Cuáles son los cuatro pasos del triaje y por qué la línea de cambios va tan pronto?
2. ¿Qué pregunta descarta cada capa y en qué orden se recorren?
3. ¿Cuándo no conviene mitigar antes de entender?

4. ¿Cómo se aborda un fallo intermitente?
5. ¿Qué pregunta suele revelar la segunda causa de un incidente?

Referencias

- Beyer, B. y otros (2016). Site Reliability Engineering, cap. «Effective troubleshooting». <https://sre.google/sre-book/effective-troubleshooting/>
- Allspaw, J. (2015). Trade-offs under pressure: heuristics and observations of teams resolving internet service outages. <https://www.researchgate.net/publication/282869185>
- Huang, P. y otros (2017). Gray failure: the Achilles' heel of cloud-scale systems. <https://www.microsoft.com/en-us/research/publication/gray-failure-the-achilles-heel-of-cloud-scale-systems/>
- Nygard, M. (2018). Release It!, 2.ª ed. — patrones de fallo y su firma. <https://pragprog.com/titles/mnee2/release-it-second-edition/>
- Google (2018). The Site Reliability Workbook: incident response. <https://sre.google/workbook/incident-response/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 21 en PDF · Manual integral

Anterior	Índice	Siguiente
← 257 · Alertas, on-call, escalamiento y comunicación	Parte 21 · Programa	259 · Runbooks ejecutables y auto-remediation →

Evaluación — 258 Triage de red, cómputo, datos y dependencias

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 diagnostic-tree 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.

259 — Runbooks ejecutables y auto-remediation

Parte: 21 — Operación cloud, automatización y respuesta a incidentes Nivel: avanzado · Horas estimadas: 4
Laboratorio: operations · Estado: EXECUTABLECORE

Propósito

Convertir procedimientos escritos en procedimientos que se ejecutan. La clase distingue los tres grados —documentado, asistido y automático—, da el criterio para subir de grado sin automatizar lo que no se debe, y monta la remediación automática con sus dos requisitos innegociables: saber cuándo NO actuar y dejar rastro de lo que hizo.

Resultados de aprendizaje

Al finalizar podrás:

1. Distinguir procedimiento documentado, asistido y automático.
2. Decidir qué se automatiza y qué se queda con humano en el bucle.
3. Escribir procedimientos que no envejecen porque se ejecutan.
4. Montar remediación automática con límites y rastro.
5. Evitar que la automatización empeore el incidente.

Conceptos centrales

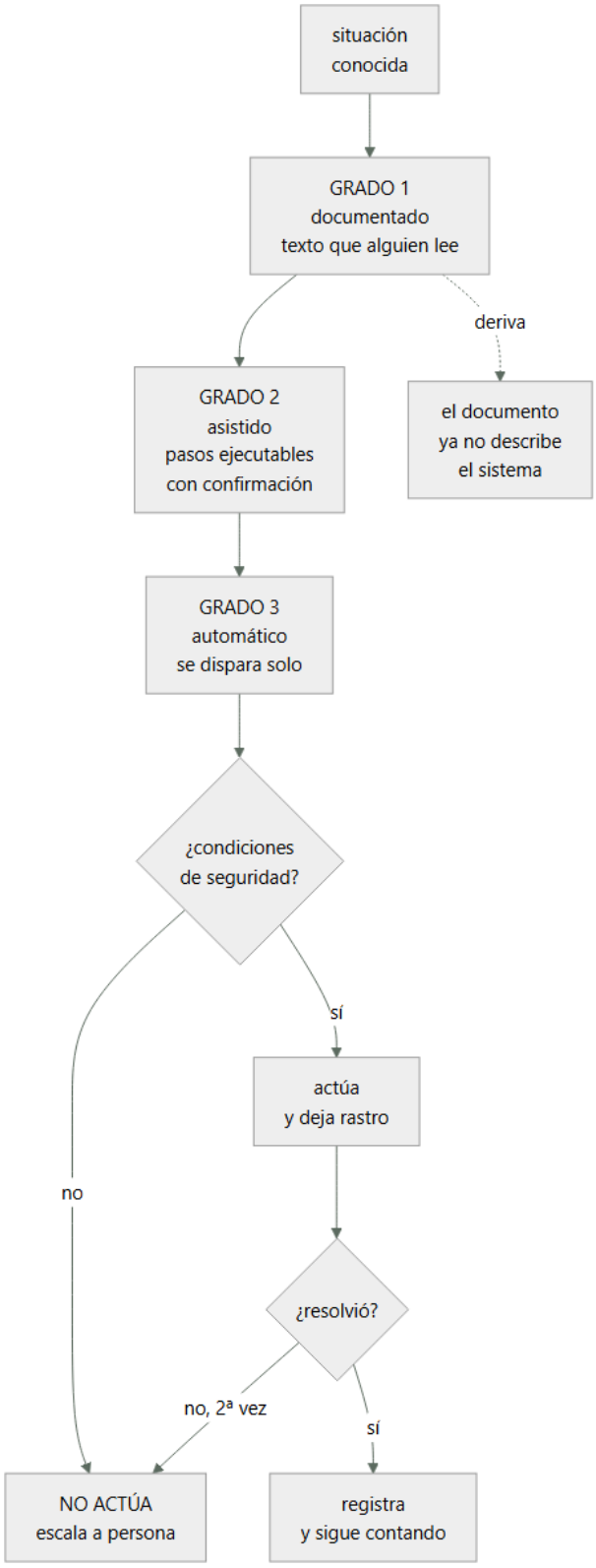
Concepto	Comprensión verificable
procedimiento	Guía de respuesta para una situación conocida. Su valor está en ser correcta cuando se usa, no en existir.
procedimiento asistido	Documento donde los pasos son ejecutables desde él, con confirmación humana.
remediación automática	Acción correctiva disparada por una señal, sin intervención humana.
límite de acción	Cuántas veces y en qué condiciones puede actuar la automatización antes de parar y pedir ayuda.
interruptor de la automatización	Forma de desactivarla al instante, sin desplegar.
deriva del procedimiento	El documento describe un sistema que ya no existe. El estado por defecto de todo procedimiento escrito.

Modelo mental

Operar es controlar cambios bajo incertidumbre: inventario, señal, diagnóstico, acción reversible y aprendizaje deben formar un ciclo cerrado.

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. Los tres grados

Un procedimiento no es un documento: es una capacidad de respuesta. Y tiene tres grados según cuánto trabajo humano exige.

```
GRADO 1 · DOCUMENTADO
  texto que alguien lee y traduce a comandos
```

```
  ventaja  barato de escribir
  problema DERIVA
           el documento envejece y nadie se entera hasta
           el incidente
           → y a las 03:00 nadie descubre que el comando
             ya no existe

GRADO 2 · ASISTIDO
  el procedimiento contiene los pasos EJECUTABLES
  → un cuaderno, un flujo de trabajo, un asistente de chat
  → cada paso se ejecuta desde el documento, con
    confirmación

  ventaja  no hay traducción ni erratas
           y si un paso se rompe, se rompe al ejecutarlo:
           la deriva se detecta
           el rastro sale gratis            clase 257
  → y es el grado que más valor aporta por el esfuerzo

GRADO 3 · AUTOMÁTICO
  una señal dispara la acción, sin persona
  ventaja  segundos en vez de minutos, y de noche igual
           que de día
  riesgo   puede empeorar el incidente
  → y por eso necesita condiciones, límites y rastro
```

Y el criterio para subir de grado:

```
de 1 a 2  SIEMPRE que el procedimiento se use más de una
          vez al trimestre
          → el coste es bajo y elimina la deriva

de 2 a 3  cuando las tres se cumplen
          a  el diagnóstico es INEQUÍVOCO
             la señal identifica la situación sin
             ambigüedad
          b  la acción es SEGURA si el diagnóstico fuera
             erróneo
             → «¿qué pasa si actúa cuando no debía?»
          c  la frecuencia lo justifica
             → automatizar lo que ocurre una vez al año
               produce automatización sin probar, que es
               peor que nada

→ y si (b) no se cumple, se queda en grado 2 aunque duela
```

Y los candidatos naturales de cada grado:

```
GRADO 3, claros
  retirar una instancia que falla la comprobación de salud
  reiniciar un proceso con fuga detectada
  ampliar un volumen que llega al 85 %
  rotar un certificado a punto de caducar   clase 196
  limpiar registros antiguos
  reintentar un flujo de datos fallido       clase 244
  revertir un despliegue cuyo canario falla  clase 102

GRADO 2, y no más
  conmutar de región                         clase 187
  restaurar desde copia                      clase 189
  cambiar el enrutado del tráfico
  desactivar una función
  todo lo que toque datos de forma no reversible

GRADO 1 basta
  lo que ocurre una vez al año y es complejo
  → pero entonces hay que ENSAYARLO         clase 261
```

2. Procedimientos que no envejecen

La deriva es el estado por defecto: el sistema cambia y el documento no.

```
LO QUE LA CAUSA
  el documento vive lejos del código
  quien cambia el sistema no sabe que existe el documento
```

```
  y nadie lo lee hasta que hay un incidente

LO QUE LA EVITA
  1  que los pasos sean EJECUTABLES
     → si el paso está roto, falla al ejecutarlo
  2  que se EJECUTEN aunque no haya incidente
     → ensayos periódicos                    clase 261
  3  que vivan con el código del servicio
     → y se revisen en el mismo cambio
  4  que la alerta ENLACE al procedimiento   clase 257
     → y si el enlace está roto, se ve en cada alerta
```

Y qué contiene un procedimiento útil:

```
ENCABEZADO
  para qué alerta / síntoma
  cuál es el impacto en el usuario
  y cuál es la acción de MITIGACIÓN inmediata
  → arriba del todo, no al final

DIAGNÓSTICO
  las tres o cuatro comprobaciones que discriminan
  con las consultas ENLAZADAS, no descritas   clase 258

ACCIÓN, por escenario
  si A → hacer X
  si B → hacer Y
  si nada encaja → escalar a quien           clase 257

Y LO QUE NO HAY QUE HACER
  → «no reinicies la base: tarda 40 minutos en recuperar»
  → esta sección evita más daño que ninguna otra

VERIFICACIÓN
  cómo se sabe que quedó arreglado
  → una señal concreta, no «comprobar que va bien»
```

Y lo que NO debe contener:

```
explicaciones de arquitectura       → en la documentación
historia del incidente que lo originó
listas de comandos sin contexto de cuándo usarlos
y nada que ocupe más de una pantalla antes de la primera
  acción útil

→ un procedimiento se lee a las 03:00, con prisa y con
  ruido
→ y el criterio de calidad es: ¿lo puede seguir alguien
  que no construyó el servicio?
→ que es exactamente lo que se prueba en el ensayo
                                              clase 261
```

3. Remediación automática, con límites

Automatizar la respuesta es lo que convierte incidentes en no-eventos. Y también lo que convierte un problema pequeño en uno grande, si falta lo siguiente.

```
REQUISITO 1 · SABER CUÁNDO NO ACTUAR
  condiciones de seguridad antes de cada acción
    ¿cuántos objetivos sanos quedan?
      → si retirar esta instancia deja menos del mínimo,
        NO retirar
    ¿es un fallo local o general?
      → si fallan TODAS, el problema no es la instancia
      → y reiniciarlas todas destruye lo que quedaba
    ¿hay un despliegue en curso?
    ¿ya actuó por lo mismo hace poco?

REQUISITO 2 · LÍMITE DE ACCIONES
  «como mucho N veces por ventana»
    → y a la N+1, no actúa: ESCALA
  → porque si la acción no resolvió dos veces, el
    diagnóstico es falso
  → y el bucle de remediación es un modo de fallo real
```

```
REQUISITO 3 · RASTRO
  qué señal la disparó, qué comprobó, qué hizo, con qué
  resultado
  → visible en el mismo sitio que los cambios humanos
                                                clase 258
  → una automatización sin rastro convierte cada incidente
    en un misterio

REQUISITO 4 · INTERRUPTOR
  desactivarla al instante, sin desplegar     clase 105
  → y quien coordina el incidente debe poder usarlo

REQUISITO 5 · QUE AVISE
  actuó, luego se cuenta
  → aviso, no alerta: nadie tiene que despertarse
                                                clase 257
  → pero si actúa 40 veces en una semana, eso SÍ es una
    alerta: hay un problema que nadie está viendo
```

Y la métrica que revela si la automatización tapa problemas:

```
CUÁNTAS VECES ACTÚA, por causa y por semana
  estable y baja      bien
  creciendo           hay un problema de fondo
  muy alta            la automatización es una tirita

→ y ese contador es lo que impide que la remediación
  automática se convierta en la forma de no arreglar nada
→ que es su mayor riesgo, y no es técnico
```

Y el escalón entre grado 2 y 3, que casi siempre conviene:

```
MODO SOMBRA
  la automatización evalúa y REGISTRA lo que HABRÍA hecho
  sin hacerlo
  → dos o cuatro semanas
  → y se revisa: ¿habría acertado siempre?

→ y aquí se descubren las condiciones que faltaban
→ es el mismo patrón de las políticas en aviso  clase 217
→ y del canario                                 clase 102
```

4. Los modos de fallo de la automatización

Automatizar la respuesta introduce fallos nuevos. Conviene conocerlos antes de sufrirlos.

```
1  EL BUCLE
   la acción dispara la señal que dispara la acción
   → escalar por CPU, y el arranque consume CPU
   → reiniciar por plazo, y el reinicio causa plazos
   defensa  límite de acciones y ventana de calma

2  LA REMEDIACIÓN QUE ESCONDE
   reinicia el proceso con fuga cada 6 horas, en silencio
   → y nadie arregla la fuga en dos años
   defensa  contador visible y revisión periódica

3  LA ACCIÓN CORRECTA EN EL MOMENTO EQUIVOCADO
   retirar instancias no sanas está bien
   → salvo durante un despliegue, donde no estar sano es
     temporal
   defensa  condición «¿hay cambio en curso?»

4  LA AUTOMATIZACIÓN QUE NO SE PROBÓ
   escrita hace 14 meses, nunca disparada
   → y cuando dispara, falla por permisos            ley 15
   defensa  ejercitarla en los ensayos          clase 261

5  LA QUE ACTÚA SOBRE UN DIAGNÓSTICO CORRELACIONAL
   «si la latencia sube, escalar»
   → y la latencia subía por una dependencia lenta
   → escalar multiplicó la carga sobre ella     clase 201
   defensa  el criterio (a): diagnóstico inequívoco

6  Y LA QUE NADIE SABE QUE EXISTE
```

```
    el incidente se comporta de forma inexplicable porque
    algo está actuando
    defensa  rastro en la línea de cambios       clase 258
    → y un inventario de automatizaciones activas
```

Y el papel de los procedimientos en el ciclo completo:

```
incidente → análisis posterior → acción             clase 111
  y la acción tiene tres formas, en orden de valor
    1  eliminar la causa
    2  automatizar la respuesta
    3  documentar la respuesta

→ y (2) sin intentar (1) es deuda
→ pero (2) mientras (1) se hace bien es correcto
→ y (3) sola, sin plazo para subir de grado, es lo que
  produce carpetas de documentos muertos
```

Y la lista de comprobación de la clase:

- cada alerta accionable enlaza a un procedimiento
- los procedimientos empiezan por impacto y mitigación
- los pasos son ejecutables, no descritos
- tienen sección de «lo que NO hay que hacer»
- tienen verificación con señal concreta
- viven con el código y se revisan en el mismo cambio
- se ejecutan en los ensayos, no solo en incidentes
- lo automatizado cumple diagnóstico inequívoco y acción segura
- toda remediación tiene condiciones de seguridad
- tiene límite de acciones por ventana y escala al superarlo
- deja rastro en la línea de cambios
- tiene interruptor sin despliegue
- se estrenó en modo sombra
- hay contador de actuaciones por causa, revisado
- existe inventario de automatizaciones activas

Y el cierre que enlaza con la clase siguiente: la mayor fuente de incidentes es el cambio, y el triaje lo confirmó: el 58 % se resolvió revirtiendo uno. Gestionar el cambio —ventanas, congelaciones y vuelta atrás— es la materia de la clase 260.

Ejemplo trabajado

CloudShop convierte 34 procedimientos en capacidad de respuesta. Lo que sigue es el inventario que encontró que 19 de 34 estaban rotos, la primera remediación automática que causó un incidente, y las cifras a los nueve meses.

Punto de partida: el inventario.

```
34 procedimientos escritos, en tres sitios distintos
  wiki del equipo            18
  repositorio de la
  plataforma                  9
  documentos sueltos          7

y la prueba: coger cada uno y EJECUTARLO en un entorno
de ensayo

     funciona íntegro                       11
     algún comando ya no existe             13
     referencia a un servicio retirado       4
     permisos insuficientes para quien
     estaría de guardia                      2
     el procedimiento describe un sistema
     que ya no es así                        4

     rotos, en total                    19/34   56 %
     antigüedad media de la última
     edición                          14 meses
```

Y el hallazgo que más molestó:

```
el procedimiento de conmutación de región
  escrito con mucho detalle, 6 páginas
```

```
  nunca ejecutado
  → el nombre del grupo de recursos secundario había
    cambiado hacía 8 meses
  → y tres de los pasos requerían un permiso que el rol de
    guardia no tenía                        clases 189, 231

→ el procedimiento del que más dependía el plan de
  continuidad era el más roto
→ ley 15, otra vez
```

La conversión, por grados.

```
de los 34
  se retiraron por obsoletos                       9
  quedaron 25

  a GRADO 2 (asistido)                            25
    cuadernos ejecutables enlazados desde la alerta
    esfuerzo medio por procedimiento         3.5 horas

  y de esos, subieron a GRADO 3 (automático)       8
```

Y los ocho que subieron, con su criterio:

```
retirar instancia no sana              sí        sí      31
reiniciar proceso con fuga             sí        sí      12
ampliar volumen al 85 %                sí        sí       6
rotar certificado < 21 días            sí        sí       4
reintentar flujo de datos fallido      sí        sí      18
revertir despliegue con canario malo   sí        sí       3
limpiar registros antiguos             sí        sí       9
liberar conexiones huérfanas           sí        sí       7
```

Y los que se quedaron en grado 2, con el motivo:

```
conmutar de región          acción no segura si el
                            diagnóstico falla
restaurar desde copia       no reversible
desviar tráfico entre       impacto amplio; decisión de
proveedores                 negocio
desactivar función de pago  decisión de negocio
ampliar cuota de la base    coste; requiere aprobación

→ el criterio (b) fue el que descartó a la mayoría
```

El incidente que causó la primera remediación automática.

```
semana 3
  la automatización de «retirar instancia no sana» actuó
  correctamente 14 veces

semana 4, martes, 09:12
  una dependencia común empieza a responder lento
  → las comprobaciones de salud de 9 de 11 instancias
    empiezan a fallar por plazo

09:12  retira una
09:12  retira otra
09:13  retira otra
  ...
09:14  quedan 2 instancias sirviendo todo el tráfico
09:14  las 2 se saturan y también fallan
09:15  servicio caído por completo

tiempo desde el primer síntoma hasta la caída total
                                            3 minutos

y sin la automatización
  el servicio habría estado LENTO, no caído
  → la automatización convirtió una degradación en una
    caída
```

Y lo que faltaba, exactamente:

```
la condición de MÍNIMO SANO
  «no retirar si quedarían menos de N sanas»
```

```
y la condición de FALLO GENERAL
  «si falla más del 50 % a la vez, el problema no son las
  instancias»
  → no actuar; escalar

y el LÍMITE POR VENTANA
  «como mucho 2 retiradas cada 10 minutos»

→ las tres se añadieron
→ y se añadió el interruptor, que en ese incidente no
  existía: hubo que desplegar para pararla, 11 minutos más
```

Y el cambio de proceso que siguió:

```
ninguna remediación entra sin
  a  4 semanas en modo sombra
  b  condiciones de seguridad revisadas por otra persona
  c  interruptor probado
  d  contador de actuaciones en el panel
  e  y un ensayo donde se dispara a propósito  clase 261

→ las 7 restantes pasaron por esto
→ y en modo sombra, 2 de las 7 revelaron condiciones que
  faltaban
    reintentar flujo de datos → habría reintentado un
      fallo por datos malos indefinidamente     clase 244
    limpiar registros → habría borrado durante una
      investigación abierta
```

Las cifras a los nueve meses.

```
procedimientos                                34          25
  ejecutables (grado 2 o 3)                    0          25
  rotos al probarlos                       19/34        1/25
  antigüedad media de la última edición   14 meses    19 días

situaciones resueltas sin persona             0/mes    73/mes
tiempo medio de respuesta a esas
  situaciones                              18 min       40 seg
interrupciones de guardia por turno           9.4         2.6
interrupciones fuera de horario por turno     3.1         0.7

incidentes causados por la automatización       -           1
  tras añadir condiciones y sombra              -           0
```

Y el contador que reveló un problema de fondo:

```
«reiniciar proceso con fuga» empezó actuando 12 veces/mes
  mes 4      19
  mes 5      31
  mes 6      47

→ la revisión del contador lo detectó
→ la fuga se había agravado con un cambio del mes 4
→ y sin el contador, la automatización lo habría escondido
  indefinidamente

se arregló la fuga
  mes 7       2
  mes 8       0
```

Y el coste, para dimensionar:

```
conversión de 25 procedimientos a grado 2   88 horas
8 remediaciones automáticas                 96 horas
modo sombra y ajustes                       34 horas
                                           ■■■■■■■■■
                                           218 horas

y el retorno, solo en interrupciones evitadas
  73 situaciones/mes × 18 min = 22 horas/mes
  → recuperado en 10 meses en tiempo directo
  → y el valor real está en las 2.4 interrupciones
    nocturnas por turno que dejaron de ocurrir
```

La lección que esta clase deja: los 34 procedimientos existían y 19 estaban rotos —incluido el de conmutación de región, el más importante— porque un documento que nunca se ejecuta no tiene forma de avisar de que ha envejecido. Y la primera remediación automática convirtió una degradación en una caída total en tres minutos, no por estar mal programada, sino por no saber cuándo no actuar.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-21-cloud-operations-automation/259-runbooks-ejecutables-y-auto-remediation/lab.py
```

El laboratorio selecciona el motor de práctica operations 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 executable-runbook 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 runbook probado por otra persona. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye executable-runbook 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 procedimiento no funciona cuando se necesita a las 03:00	Es texto que nadie ha ejecutado desde que se escribió	Conviértelo en pasos ejecutables y ejercítalo en los ensayos; la deriva solo se detecta ejecutando.
La remediación automática convirtió una degradación en una caída	No tenía condición de mínimo sano ni detección de fallo general	Antes de cada acción comprueba cuántos objetivos sanos quedan y si el fallo es local o general; si es general, no actúes y escala.
Una automatización actúa sin parar y nadie arregla la causa	No hay contador visible ni revisión de cuántas veces actúa	Mide actuaciones por causa y semana, y revísalo; si crece, hay un problema de fondo que la automatización tapa.
Durante un incidente el sistema hace cosas que nadie ordenó	La remediación no deja rastro ni hay inventario de lo que está activo	Registra cada actuación en la línea de cambios y manten un inventario de automatizaciones con su interruptor.
Hubo que desplegar para parar la automatización durante el incidente	No existía interruptor independiente del despliegue	Toda remediación necesita desactivarse al instante mediante una bandera, y quien coordina debe poder usarla.
Se automatizó algo que se usa una vez al año y falló al dispararse	Frecuencia insuficiente para que la automatización esté probada	Si ocurre raramente, quédate en grado 2 y ensáyalo; la automatización sin uso es automatización sin probar.

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. ¿Cuáles son los tres grados y qué criterio permite subir de 2 a 3?
2. ¿Qué hace que un procedimiento envejezca y qué lo evita?
3. ¿Qué condiciones de seguridad necesita toda remediación automática?
4. ¿Para qué sirve el modo sombra y qué suele revelar?
5. ¿Qué indica que una automatización está tapando un problema en vez de resolverlo?

Referencias

- Beyer, B. y otros (2018). The Site Reliability Workbook, cap. «Automation and toil». <https://sre.google/workbook/eliminating-toil/>
- AWS (2024). Systems Manager Automation runbooks. <https://docs.aws.amazon.com/systems-manager/latest/userguide/automation-documents.html>
- Microsoft (2024). Azure Automation runbooks. <https://learn.microsoft.com/azure/automation/automation-runbook-types>
- Google Cloud (2024). Automated remediation with Cloud Functions and alerting. <https://cloud.google.com/architecture/automated-remediation-cloud-monitoring>
- Limoncelli, T. (2020). Operational excellence in April Fools' pranks — sobre automatización probada frente a escrita. <https://queue.acm.org/detail.cfm?id=3380780>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 21 en PDF · Manual integral

Evaluación — 259 Runbooks ejecutables y auto-remediation

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 executable-runbook 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.

260 — Change management, ventanas y rollback

Parte: 21 — Operación cloud, automatización y respuesta a incidentes Nivel: avanzado · Horas estimadas: 4
Laboratorio: delivery · Estado: EXECUTABLECORE

Propósito

Gestionar el cambio sin frenarlo. La clase demuestra que cambiar más despacio no reduce el riesgo: lo concentra, explica por qué los procesos de aprobación pesados empeoran las cifras, y da el conjunto que sí funciona: cambios pequeños y frecuentes, vuelta atrás probada, ventanas y congelaciones con criterio, y clasificación por riesgo real.

Resultados de aprendizaje

Al finalizar podrás:

1. Explicar por qué frenar el cambio concentra el riesgo en vez de reducirlo.
2. Clasificar los cambios por riesgo real, no por burocracia.
3. Diseñar vuelta atrás que funcione, incluida la que no es reversible.
4. Aplicar ventanas y congelaciones con criterio y con coste explícito.
5. Medir el proceso de cambio con las cuatro métricas que importan.

Conceptos centrales

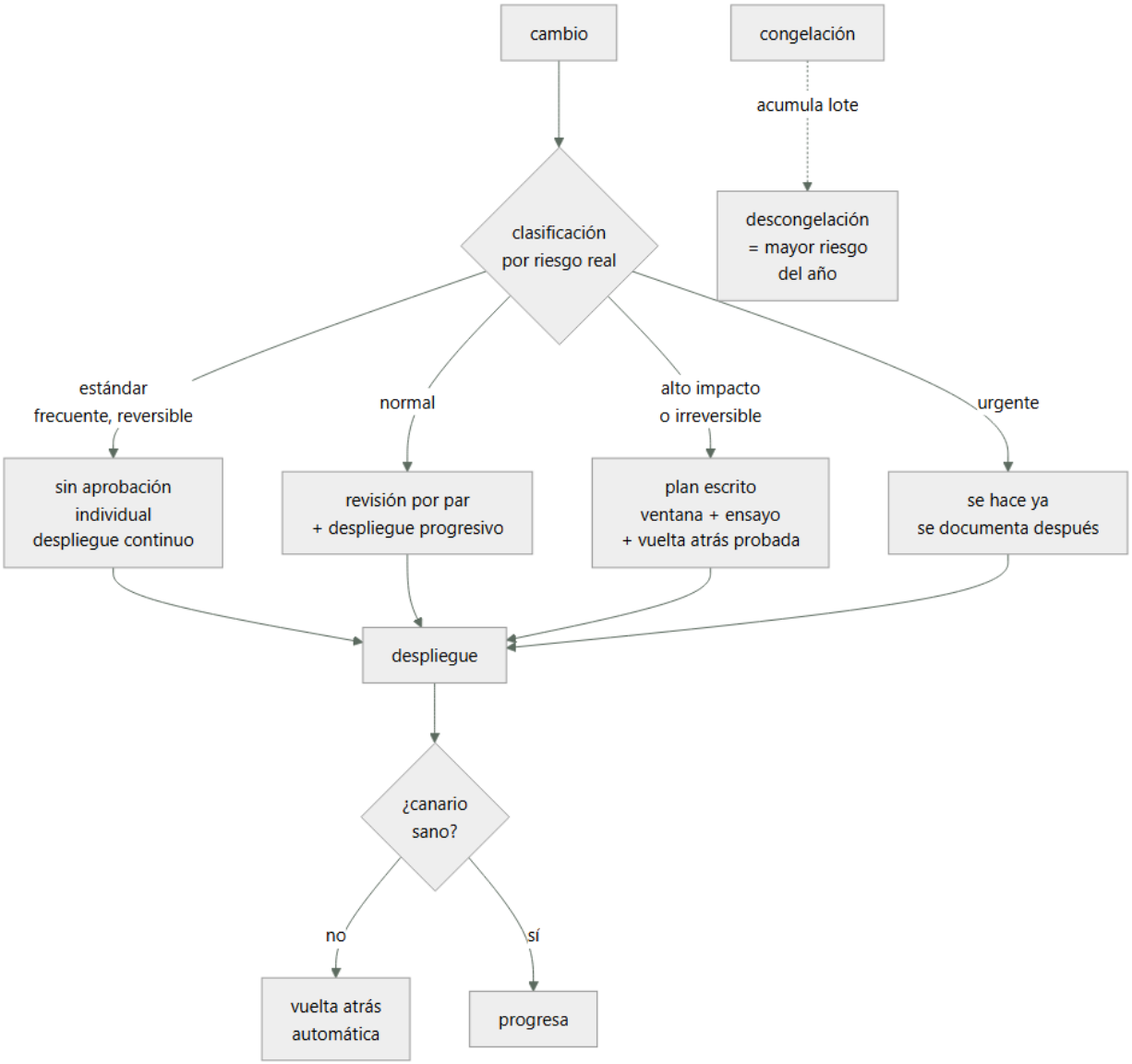
Concepto	Comprensión verificable
lote	Cuántos cambios entran juntos. La variable que más determina el riesgo de un despliegue.
cambio estándar	Cambio frecuente, de riesgo conocido y con vuelta atrás probada. No necesita aprobación individual.
vuelta atrás	Devolver el sistema al estado anterior. Solo cuenta si se ha probado.
avance hacia delante	Arreglar aplicando un cambio nuevo. La única salida cuando la vuelta atrás no es posible.
congelación	Periodo sin cambios. Traslada el riesgo al momento en que se descongela.
ventana de cambio	Franja horaria en que se permite desplegar. Útil cuando el impacto depende de la hora, dañina cuando no.

Modelo mental

Operar es controlar cambios bajo incertidumbre: inventario, señal, diagnóstico, acción reversible y aprendizaje deben formar un ciclo cerrado.

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. Frenar el cambio no reduce el riesgo

La intuición dice que menos cambios significan menos incidentes. Los datos dicen lo contrario, y la razón es aritmética.

```
EL RIESGO DE UN DESPLIEGUE NO ES CONSTANTE:
CRECE CON EL LOTE

  1 cambio      1 causa posible
                y si falla, se sabe cuál es

  40 cambios    40 causas posibles
                y las interacciones entre ellas
                → y el diagnóstico se multiplica
                → y la vuelta atrás revierte 39 cambios
                  buenos con el malo

→ desplegar la mitad de veces no significa la mitad de
  riesgo: significa lotes del doble de tamaño
→ y el riesgo por despliegue crece más que linealmente
```

Y lo que muestran las mediciones del sector:

```
los equipos que despliegan MÁS a menudo tienen
  menor tasa de fallo por cambio
  menor tiempo de recuperación
```

```
  y menor tiempo de entrega

→ no a pesar de desplegar más, sino PORQUE despliegan más
→ frecuencia y estabilidad correlacionan POSITIVAMENTE
→ es el hallazgo más contraintuitivo y mejor replicado
  del campo
```

Y por qué las aprobaciones pesadas empeoran las cifras:

```
un comité que aprueba cambios
  no conoce el cambio mejor que quien lo escribió
  → y por tanto no puede evaluar su riesgo real
  añade días de espera
  → y la espera acumula lote
  y desplaza la responsabilidad
  → «lo aprobó el comité» sustituye a «lo comprobé»

→ el efecto medido: peor tiempo de entrega Y peor tasa de
  fallo
→ la aprobación externa funciona peor que la revisión por
  un par                                          clase 102
```

Y lo que sí reduce el riesgo:

```
lote pequeño
despliegue progresivo con canario           clase 102
vuelta atrás probada
separación de despliegue y activación       clase 105
automatización de la comprobación           clase 100
y revisión por alguien que entiende el cambio

→ ninguna de estas es un proceso de aprobación
→ todas son propiedades del sistema de entrega
```

2. Clasificar por riesgo real

No todos los cambios necesitan el mismo tratamiento. La clasificación útil se hace por riesgo, no por tipo de recurso.

```
ESTÁNDAR
  frecuente, con vuelta atrás probada, impacto acotado
  → despliegue de código con canario
  → cambio de configuración con bandera
  → escalado dentro de límites conocidos
  aprobación  ninguna individual; se aprobó el
              PROCEDIMIENTO, no el cambio
  → y aquí debe caer el 90 % o más de los cambios

NORMAL
  no rutinario, reversible, impacto conocido
  → revisión por un par que entienda el área
  → despliegue progresivo

ALTO IMPACTO O IRREVERSIBLE
  migración de datos, cambio de esquema destructivo,
  cambio de red troncal, rotación de clave maestra
  → plan escrito con pasos y verificación
  → ensayo previo en entorno equivalente
  → ventana acordada
  → vuelta atrás PROBADA, o plan de avance hacia delante
  → y una persona designada que puede detenerlo

URGENTE
  se hace ya, con quien coordina el incidente enterado
  → y se documenta DESPUÉS, sin excepción
  → el registro posterior es lo que evita que «urgente»
    se convierta en la vía de escape habitual

→ y la métrica de salud del proceso es qué porcentaje de
  cambios se declara urgente
→ si pasa del 5 %, la clasificación normal es demasiado
  lenta y la gente la rodea
```

Y la trampa clásica de la clasificación:

```
clasificar por RECURSO en vez de por RIESGO
```

```
  «todo cambio en producción es alto impacto»
  → y entonces todo pasa por el proceso pesado
  → y el proceso pesado se convierte en un trámite que
    nadie lee

  «cambiar una variable de entorno es trivial»
  → y la variable era la cadena de conexión
                                        clase 232

→ el riesgo lo determina el ALCANCE DEL FALLO POSIBLE y
  la REVERSIBILIDAD, no el tipo de recurso
```

3. Vuelta atrás que funciona

«Podemos revertir» es la frase más repetida y menos comprobada de la operación.

```
CUÁNDO LA VUELTA ATRÁS ES REAL
  el artefacto anterior sigue existiendo
  el procedimiento se ha ejecutado en los últimos 30 días
  el tiempo de vuelta atrás se conoce, medido
  y no depende de nada que el fallo haya roto

CUÁNDO NO LO ES
  cambio de esquema que ya escribió datos nuevos
  migración de datos ya iniciada
  mensajes ya consumidos con formato nuevo   clase 210
  claves rotadas y las anteriores destruidas  clase 197
  caché envenenada que sobrevive al despliegue
  y cambios de terceros, que no revierten cuando tú
    reviertes

→ y en todos esos casos, la salida es AVANZAR
→ y eso hay que decidirlo ANTES, no durante
```

Y el patrón que hace reversible lo que no lo parece:

```
CAMBIO DE ESQUEMA EN CUATRO PASOS       clase 209
  1  añadir lo nuevo sin quitar lo viejo
  2  escribir en ambos
  3  leer de lo nuevo
  4  y solo entonces, retirar lo viejo

→ cada paso es reversible por separado
→ el paso 4 se hace días o semanas después
→ y es lo que convierte una migración irreversible en
  cuatro cambios estándar

y el mismo patrón vale para
  formatos de mensaje                    clase 210
  contratos de interfaz                  clase 106
  y particiones de datos                 clase 208
```

Y la separación que más margen da:

```
DESPLEGAR ≠ ACTIVAR                      clase 105
  el código va a producción apagado
  → y se enciende con una bandera, por porcentaje
  → y se apaga en segundos, sin desplegar

→ la vuelta atrás de una bandera es instantánea
→ la de un despliegue tarda minutos
→ y en un incidente, esa diferencia es todo
```

Y qué hacer con el cambio que no se puede probar en pequeño:

```
un cambio de red troncal, una rotación de clave maestra
  → no hay canario posible
  la defensa es
    ensayo en entorno equivalente        clase 261
    ventana con la gente disponible
    verificación paso a paso, no al final
    punto de no retorno EXPLÍCITO y anunciado
    y criterio de abandono escrito ANTES
      → «si a los 20 minutos X no ocurre, abortamos»
```

4. Ventanas, congelaciones y su coste

Ventanas y congelaciones son herramientas legítimas mal usadas casi siempre.

```
VENTANA DE CAMBIO
  útil cuando el impacto DEPENDE DE LA HORA
    → un sistema de pago con pico a mediodía
    → un proceso por lotes nocturno
  dañina cuando no
    → «solo se despliega los martes de 14 a 16»
    → acumula lote, concentra el riesgo y multiplica la
      espera

  y la trampa nocturna
    desplegar de madrugada suena prudente
    → pero hay menos gente, más cansada, y menos tráfico
      para detectar el problema
    → y el fallo se descubre a las 09:00 con el equipo
      entrando
  → desplegar cuando hay gente disponible y tráfico
    suficiente para ver el efecto es MEJOR

CONGELACIÓN
  legítima cuando
    hay un evento de negocio con impacto extremo
      → temporada alta, lanzamiento, cierre fiscal
    y dura poco

  y su coste, que casi nunca se calcula
    el lote acumulado se libera de golpe al descongelar
    → el día de la descongelación es el de mayor riesgo
      del año
    los arreglos de seguridad también se congelan
    → y esa es la parte peligrosa            clase 216
    y el equipo pierde la práctica de desplegar

  cómo hacerla bien
    excepción explícita para seguridad y para incidentes
    descongelación ESCALONADA, no de golpe
    y medir cuántos incidentes ocurren en la semana
    siguiente
```

Y las cuatro métricas que dicen si el proceso de cambio es sano:

```
FRECUENCIA DE DESPLIEGUE
  → cuántas veces se entrega valor
TIEMPO DE ENTREGA
  → de confirmar el cambio a estar en producción
TASA DE FALLO POR CAMBIO
  → qué porcentaje causa degradación
TIEMPO DE RECUPERACIÓN
  → cuánto se tarda en restablecer

→ y las cuatro mejoran juntas o empeoran juntas
→ un proceso que mejora la tasa de fallo empeorando la
  frecuencia NO está mejorando
→ y esa es la comprobación que desenmascara los procesos
  de aprobación pesados
```

Y la lista de comprobación de la clase:

- ■ el 90 % o más de los cambios son estándar
- ■ los cambios estándar no requieren aprobación individual
- ■ la revisión la hace quien entiende el área, no un comité
- ■ menos del 5 % de los cambios se declaran urgentes
- ■ los urgentes se documentan después, siempre
- ■ la clasificación es por alcance y reversibilidad
- ■ el despliegue es progresivo con canario y vuelta atrás automática
- ■ la vuelta atrás se ha ejecutado en los últimos 30 días
- ■ los cambios irreversibles se descomponen en pasos reversibles
- ■ desplegar y activar están separados
- ■ los cambios de alto impacto tienen criterio de abandono

```
  escrito antes
■ las ventanas existen solo donde el impacto depende de
  la hora
■ no se despliega de madrugada por costumbre
■ las congelaciones tienen excepción para seguridad
■ la descongelación es escalonada
■ se miden las cuatro métricas y se miran juntas
```

Y el cierre que enlaza con la clase siguiente: todo lo montado hasta aquí —guardia, triaje, procedimientos, vuelta atrás— vale lo que valga cuando se usa de verdad. Provocar los fallos a propósito, en horario laboral y con la gente delante, es la materia de la clase 261.

Ejemplo trabajado

CloudShop desmonta su comité de cambios. Lo que sigue son los datos que lo justificaron, la congelación de temporada alta que costó más de lo que ahorró, y las cuatro métricas antes y después.

El punto de partida: el comité.

```
comité de cambios, semanal, miércoles 10:00
  7 personas, 90 minutos
  todo cambio a producción pasaba por él

cifras de 12 meses
  cambios presentados                          1.847
  rechazados                                      11    0.6 %
  aprobados con condiciones                       23    1.2 %
  aprobados sin cambios                        1.813   98.2 %

  espera media desde listo hasta aprobado    4.2 días
  coste del comité                       546 h/persona/año
```

Y los 11 rechazos, examinados uno por uno:

```
  rechazados por falta de plan de vuelta atrás      6
  rechazados por coincidir con otro cambio          3
  rechazados por no entenderse la descripción       2

→ los 6 primeros los habría detectado una comprobación
  automática
→ los 3 siguientes, un calendario compartido
→ y los 2 últimos no eran un problema de riesgo

→ ningún rechazo se debió a que el comité detectara un
  riesgo técnico que el autor no viera
```

Y el dato que cerró la discusión:

```
incidentes causados por cambio, en esos 12 meses     34
  de cambios APROBADOS por el comité                 31
  de cambios urgentes que lo saltaron                 3

→ el 91 % de los incidentes por cambio venían de cambios
  que el comité había aprobado
→ la tasa de fallo de lo aprobado (1.7 %) era
  indistinguible de la de lo no aprobado
```

Lo que se puso en su lugar.

```
CLASIFICACIÓN
  estándar        94 % de los cambios
    revisión por un par + canario + vuelta atrás
      automática
    sin aprobación individual
  normal           5 %
    revisión por par del área + despliegue progresivo
  alto impacto     1 %
    plan escrito, ensayo, ventana, persona que puede
      detenerlo
  urgente          registrado aparte

Y COMPROBACIONES AUTOMÁTICAS que sustituyen al comité
  ¿hay procedimiento de vuelta atrás y se probó?
  ¿el canario está configurado?
```

```
  ¿coincide con otro cambio en el mismo servicio?
  ¿coincide con una ventana de negocio?
  ¿toca algo marcado como crítico?
  → y si algo falla, el cambio no sale; sin reunión

EL COMITÉ se convirtió en
  revisión MENSUAL de los cambios de alto impacto
  previstos y de los incidentes por cambio del mes
  → 90 minutos al mes, 4 personas
```

La congelación de temporada alta.

```
política heredada
  congelación total del 15 de noviembre al 5 de enero
  51 días

lo que ocurría
  cambios acumulados al descongelar          287
  incidentes en la semana posterior a la
  descongelación                               9
  incidentes en una semana normal            1.4

  → 6.4 veces la tasa normal
  → y todos los años igual, durante tres años
```

Y el coste oculto que nadie había contado:

```
durante la congelación de ese año
  arreglos de seguridad retenidos                      14
    de ellos, de severidad alta                         4
    días medios de exposición añadida            37 días
                                                clase 216
  cambios de capacidad retenidos                        6
    → y dos incidentes de temporada alta fueron por
      capacidad que se había pedido y no se aplicó
```

Y la política nueva:

```
en vez de congelación total
  cambios estándar          SIGUEN, con canario más lento
                            y ventana ampliada de
                            observación
  cambios normales          requieren aprobación del
                            responsable del servicio
  alto impacto              suspendidos, salvo excepción
                            escrita
  seguridad y capacidad     NUNCA se congelan

y el resultado de la temporada siguiente
  cambios acumulados al «descongelar»            19
  incidentes en la semana posterior             1.7
  incidentes durante la temporada alta
    año anterior (congelado)                      11
    este año                                       6

→ menos incidentes DURANTE la temporada, con cambios
  fluyendo
→ el mecanismo: los problemas de capacidad y de
  configuración se corrigieron cuando aparecieron, en vez
  de acumularse
```

Un cambio de alto impacto, con su plan.

```
migración del esquema de pedidos: partición por cliente
→ irreversible una vez escritos datos nuevos  clase 208

en vez de un cambio, CUATRO
  semana 1  añadir la clave nueva sin usarla
            → reversible; 0 riesgo
  semana 2  escribir en ambas claves
            → reversible; se compara consistencia 7 días
  semana 4  leer de la clave nueva, con bandera al 1 %,
            luego 10 %, 50 %, 100 %
            → reversible en segundos       clase 105
  semana 7  retirar la clave vieja
```

```
            → punto de no retorno, con copia verificada
                                          clase 189

y el criterio de abandono, escrito antes
  «si la comparación de consistencia muestra más de 1 por
  100.000 divergencias, se detiene y se investiga»

lo que pasó
  semana 2, día 3: 41 divergencias por 100.000
  → se detuvo
  → causa: un camino de escritura antiguo no actualizado
  → dos semanas de retraso, cero impacto en usuarios

→ y con la migración como un solo cambio, esas 41
  divergencias por 100.000 habrían llegado a producción
  como corrupción silenciosa                    ley 29
```

Las cuatro métricas, a los 14 meses.

```
frecuencia de despliegue              3.1/semana   47/semana
tiempo de entrega (confirmar → prod)     6.4 días      3.2 h
tasa de fallo por cambio                  1.8 %       0.6 %
tiempo de recuperación                   47 min      11 min

cambios por lote (mediana)                   14           1
cambios declarados urgentes                 12 %        1.9 %
incidentes causados por cambio           34/año      13/año
  con más de una causa posible
  identificada                           26/34        2/13

coste de gobierno del cambio        546 h/año     72 h/año
```

Y la cifra que mejor resume el mecanismo:

```
incidentes en que NO se supo qué cambio lo causó
  antes    26 de 34    (lotes de 14 cambios)
  después   2 de 13    (lotes de 1)

→ y el tiempo de recuperación cayó de 47 a 11 minutos
  casi todo por esto
→ revertir un cambio es rápido; encontrar cuál de catorce
  no lo es                                    clase 258
```

La lección que esta clase deja: el comité aprobó el 98,2 % de lo que le llegó y el 91 % de los incidentes por cambio venían de cambios que había aprobado; su efecto real no era filtrar riesgo sino añadir 4,2 días de espera que engordaban los lotes. Y la congelación de 51 días producía 6,4 veces la tasa normal de incidentes el día de descongelarla, mientras retenía cuatro arreglos de seguridad graves durante 37 días.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-21-cloud-operations-automation/260-change-management-ventanas-y-rollback/lab.py
```

El laboratorio selecciona el motor de práctica delivery 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 change-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 un pipeline con gates, promoción y rollback. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye change-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 despliega menos y los incidentes por cambio no bajan	Menos despliegues significan lotes mayores; el riesgo se concentra en lugar de reducirse	Reduce el tamaño del lote, no la frecuencia; un cambio por despliegue hace que revertir y diagnosticar sean triviales.
El comité aprueba casi todo y aun así hay incidentes por cambio	Quien aprueba no conoce el cambio mejor que quien lo escribió	Sustituye la aprobación por comprobaciones automáticas y revisión por un par del área; deja el comité para el 1 % de alto impacto.
La vuelta atrás falla justo cuando se necesita	Nunca se ejecutó, o el cambio ya escribió datos nuevos	Ejecuta la vuelta atrás al menos cada 30 días y descompón los cambios irreversibles en pasos reversibles.
El día de la descongelación es el peor del año	La congelación acumula lote y lo libera de golpe	Descongela de forma escalonada, y nunca congeles arreglos de seguridad ni de capacidad.
Muchos cambios se declaran urgentes para saltarse el proceso	El proceso normal es demasiado lento y la gente lo rodea	Si más del 5 % son urgentes, arregla el proceso normal; y documenta siempre los urgentes después.
Se despliega de madrugada y el fallo se descubre por la mañana	Hay menos gente, más cansada, y poco tráfico para detectar el problema	Despliega cuando hay gente disponible y tráfico suficiente para ver el efecto, salvo que el impacto dependa de la hora.

■ 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 desplegar menos a menudo no reduce el riesgo?
2. ¿Qué distingue un cambio estándar de uno de alto impacto?
3. ¿Cuándo una vuelta atrás no es posible y qué se hace entonces?
4. ¿Qué coste tiene una congelación y cómo se mitiga?
5. ¿Cuáles son las cuatro métricas del proceso de cambio y por qué hay que mirarlas juntas?

Referencias

- Forsgren, N., Humble, J. y Kim, G. (2018). Accelerate — frecuencia y estabilidad correlacionan positivamente. <https://itrevolution.com/product/accelerate/>
- Google (2024). DORA: capabilities — change approval processes. <https://dora.dev/capabilities/streamlining-change-approval/>
- Humble, J. y Farley, D. (2010). Continuous Delivery, cap. sobre gestión del cambio. <https://www.pearson.com/en-us/subject-catalog/p/continuous-delivery/P200000009415>
- AWS (2024). Operational Excellence Pillar: managing change. <https://docs.aws.amazon.com/wellarchitected/latest/operational-excellence-pillar/welcome.html>
- Microsoft (2024). Azure Well-Architected Framework: safe deployment practices. <https://learn.microsoft.com/azure/well-architected/operational-excellence/safe-deployments>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 21 en PDF · Manual integral

Anterior	Índice	Siguiente
← 259 · Runbooks ejecutables y auto-remediation	Parte 21 · Programa	261 · Game days, chaos engineering y aprendizaje →

Evaluación — 260 Change management, ventanas y rollback

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 change-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.

261 — Game days, chaos engineering y aprendizaje

Parte: 21 — Operación cloud, automatización y respuesta a incidentes Nivel: avanzado · Horas estimadas: 4
Laboratorio: chaos · Estado: EXECUTABLECORE

Propósito

Provocar los fallos a propósito para descubrir lo que no funciona antes de que lo descubra un incidente. La clase da el método de los ensayos —hipótesis, radio de impacto, criterio de parada—, la progresión de experimentos que no asusta a nadie, y la parte que casi siempre importa más que la técnica: que lo que falla en un ensayo son las personas, los procedimientos y la información, no las máquinas.

Resultados de aprendizaje

Al finalizar podrás:

1. Diseñar un ensayo con hipótesis, radio de impacto y criterio de parada.
2. Progresar desde el ensayo de mesa hasta el experimento en producción.
3. Elegir qué fallos provocar según lo que se quiere aprender.
4. Extraer de cada ensayo acciones concretas, no una sensación.
5. Justificar el ejercicio ante quien teme romper producción.

Conceptos centrales

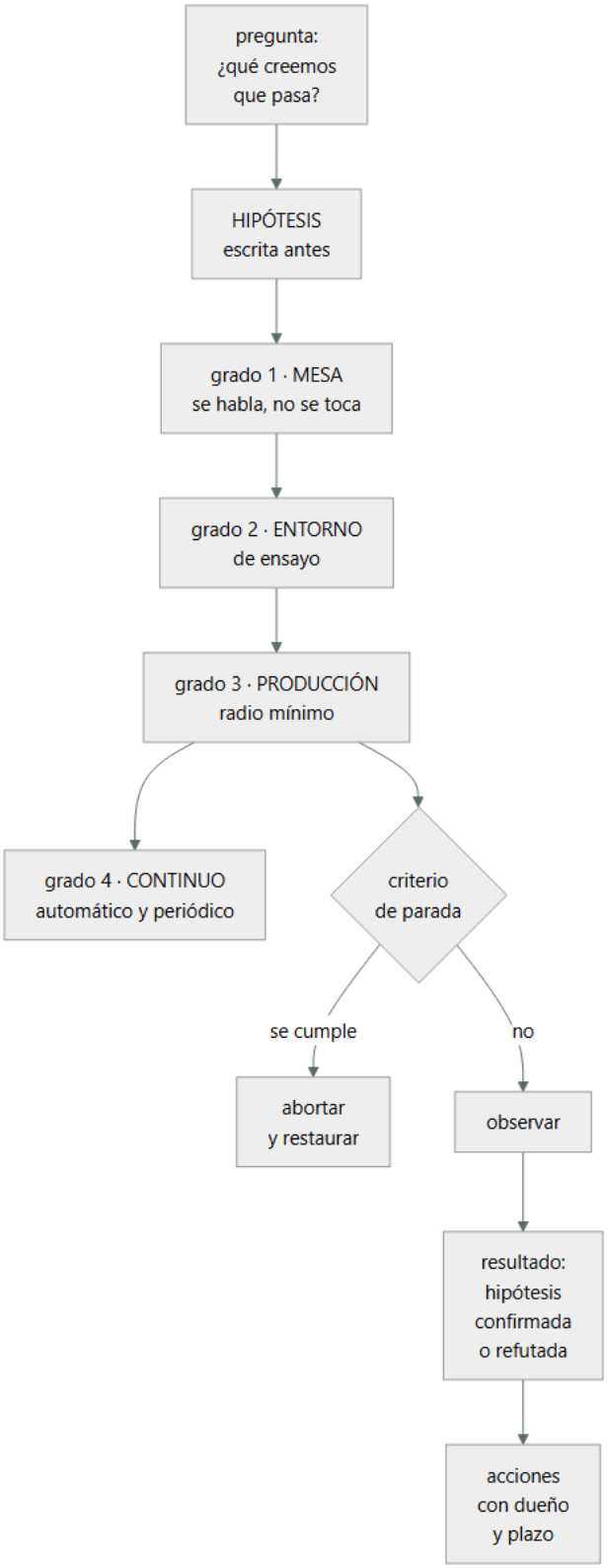
Concepto	Comprensión verificable
ensayo	Ejercicio planificado en que se provoca un fallo para observar la respuesta completa: sistema, procedimientos y personas.
hipótesis del ensayo	Lo que se espera que ocurra, escrito antes. Sin ella el ejercicio no puede fallar y no enseña nada.
radio de impacto	Cuánto puede afectar el experimento como máximo. Se empieza mínimo y se amplía con evidencia.
criterio de parada	Condición escrita ante la cual el ejercicio se aborta de inmediato.
ensayo de mesa	Se recorre el escenario hablando, sin tocar nada. El grado más barato y el que más deriva descubre.
experimento continuo	Inyección de fallos automatizada y periódica, no un evento anual.

Modelo mental

Operar es controlar cambios bajo incertidumbre: inventario, señal, diagnóstico, acción reversible y aprendizaje deben formar un ciclo cerrado.

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 se está probando de verdad

El nombre engaña. Un ensayo no prueba si el sistema aguanta: prueba si la organización responde.

```
LO QUE FALLA EN LOS ENSAYOS, por frecuencia
  1  la información no está donde debería
     el panel no muestra lo que hace falta
```

```
      la alerta no saltó, o saltó tarde
      nadie sabe dónde mirar

   2  el procedimiento está roto o no existe
      → el 56 % de CloudShop                clase 259

   3  los permisos no alcanzan
      quien está de guardia no puede ejecutar el paso 4

   4  la coordinación
      tres personas haciendo lo mismo, nadie comunicando
                                            clase 257

   5  y solo entonces, el sistema
      el mecanismo de recuperación no funciona como se creía

→ los cuatro primeros son organizativos
→ y son los que más tiempo cuestan en un incidente real
```

Y por eso el ejercicio más barato es el más rentable:

```
ENSAYO DE MESA
  una hora, una sala, un escenario
  quien lo dirige va describiendo lo que ocurre
  y los participantes dicen QUÉ HARÍAN y LO ENSEÑAN
    → «abro este panel» → que lo abra
    → «miro esta métrica» → que la enseñe
    → «ejecuto este paso» → que abra el procedimiento

→ no se toca nada, y aun así descubre 1, 2, 3 y 4
→ coste: 6 personas × 1 hora
→ y es donde debe empezar cualquier equipo
```

Y la regla que hace que el ensayo enseñe:

```
HIPÓTESIS ESCRITA ANTES
  «cuando caiga la zona A, esperamos que
     el tráfico se redistribuya en menos de 60 s
     la latencia suba menos de 30 ms
     no haya errores visibles para el usuario
     y salte la alerta de instancias sanas»

→ y sin esto, el ensayo se convierte en «hemos roto algo
  y lo hemos arreglado»
→ que no es aprendizaje: es una anécdota
→ es la misma exigencia que este programa lleva
  aplicándose desde la parte 1
```

2. El método

Un ensayo tiene cinco partes y ninguna es opcional.

```
1  PREGUNTA
   qué queremos saber que hoy no sabemos
   → y viene de un riesgo real, no de una lista genérica
   → «¿qué pasa si el proveedor de identidad no responde?»

2  HIPÓTESIS
   lo que esperamos, con números y con señales concretas
   → incluida la parte de detección: ¿saltará la alerta?

3  RADIO DE IMPACTO
   el máximo daño posible si todo sale mal
   → una instancia, no una zona
   → el 1 % del tráfico, no el 100 %
   → y se amplía SOLO con evidencia del anterior

4  CRITERIO DE PARADA
   escrito, medible, y decidido antes
   → «si los errores del usuario superan el 0,5 %,
     abortamos»
   → y quién puede declararlo: cualquiera

5  RESTAURACIÓN
   cómo se deshace, probado antes de empezar
```

```
   → y el interruptor del propio experimento
                                          clase 259
```

Y la logística que la gente subestima:

```
ANUNCIARLO
  a todos los equipos que puedan verlo
  → un ensayo sorpresa produce un incidente real con
    gente confundida
  → y quema la confianza que el ejercicio necesita

EN HORARIO LABORAL
  con la gente delante, descansada y disponible
  → hacerlo de noche contradice el propósito

CON OBSERVADORES
  alguien que solo toma notas de lo que ocurre y CUÁNDO
  → incluidas las cosas que no se dicen: dudas,
    búsquedas, sitios donde se mira y no hay nada

Y CON LA GUARDIA REAL
  el ensayo lo responde quien estaría de guardia
  → no quien construyó el sistema
  → esa diferencia es la mitad del valor
```

Y la progresión de grados, que es lo que permite empezar sin permiso especial:

```
GRADO 1  mesa                        riesgo cero
GRADO 2  entorno de ensayo           riesgo cero
         → y aquí se descubre que el entorno de ensayo no
           se parece a producción, que ya es un hallazgo
GRADO 3  producción, radio mínimo    riesgo acotado
         una instancia, un 1 %, una dependencia
GRADO 4  producción, radio amplio    riesgo real
         una zona entera, una región
GRADO 5  continuo y automático
         → inyección periódica, sin evento

→ y no se salta de grado sin haber tenido éxito en el
  anterior
→ la mayoría de las organizaciones sacan casi todo el
  valor de los grados 1 a 3
```

3. Qué fallos provocar

La elección debe venir de los riesgos del sistema, no de un catálogo. Pero el catálogo ayuda a no olvidarse.

```
DE RECURSO
  matar una instancia                    clase 213
  agotar CPU, memoria o disco
  agotar un grupo de conexiones          clase 207
  llenar un volumen

DE RED, que son los más reveladores      clase 202
  latencia añadida a una dependencia
  → y esto encuentra fallos grises que nada más encuentra
    plazos mal configurados              clase 201
    reintentos sin límite
    ausencia de cortacircuitos
  pérdida de paquetes
  bloqueo de un destino concreto
  y partición entre zonas

DE DEPENDENCIA                           clase 185
  una dependencia devuelve errores
  una dependencia responde LENTO
    → casi siempre peor que si fallara
  el proveedor de identidad no responde  clase 209
  el registro de imágenes no responde
  y el gestor de secretos no responde    clase 197

DE ZONA Y REGIÓN                         clase 187
  retirar una zona
  conmutar de región de verdad, con reloj
```

```
DE DATOS                                    clase 243
  un flujo se para
  un esquema cambia
  llegan datos con una distribución distinta

Y DE PERSONAS, los más incómodos
  quien más sabe del servicio NO participa
  → «está de vacaciones»
  el panel principal no está disponible
  el canal de chat habitual no funciona
  → y esto prueba la comunicación de verdad  clase 257
```

Y el que más sorpresas da y menos se hace:

```
LATENCIA AÑADIDA, no fallo
  200 ms extra en una dependencia

lo que suele aparecer
  plazos que eran mayores que el plazo del llamante
  reintentos que triplican la carga             clase 201
  grupos de conexiones que se agotan
  colas que crecen sin límite
  y una degradación que ninguna alerta detecta

→ porque los sistemas se prueban contra el fallo y casi
  nunca contra la LENTITUD
→ y la lentitud es la forma real en que fallan
                                        clase 185, ley 21
```

4. Del ensayo a la acción

Un ensayo sin acciones cerradas es teatro. Lo que lo convierte en aprendizaje:

```
AL TERMINAR, EN CALIENTE (30 min)
  qué esperábamos y qué pasó
  qué nos sorprendió
  qué buscamos y no encontramos
  → esta última pregunta produce la mitad de las acciones
  qué habría pasado con radio mayor

Y DE AHÍ SALEN ACCIONES
  con dueño, con plazo y con seguimiento    clase 111
  y priorizadas por lo que habrían costado en un
  incidente real

Y LA COMPROBACIÓN DE QUE SIRVIÓ
  REPETIR EL MISMO ENSAYO en 8-12 semanas
  → y la hipótesis ahora debería cumplirse
  → y si no se cumple, las acciones no funcionaron

→ repetir es lo que distingue un programa de ensayos de
  una sucesión de eventos
```

Y cómo convencer a quien teme el ejercicio, que es la parte política:

```
LA OBJECIÓN  «no vamos a romper producción a propósito»
LA RESPUESTA
  producción se rompe sola, a las 03:00, sin observadores
  y sin la gente que sabe
  → el ensayo es lo mismo con todo a favor
  → y el radio de impacto de un ensayo es MENOR que el de
    un incidente

Y CÓMO EMPEZAR SIN PEDIR PERMISO
  el ensayo de mesa no toca nada
  → y produce hallazgos suficientes para justificar el
    grado siguiente
  → CloudShop encontró 19 procedimientos rotos así
                                          clase 259

Y LA MÉTRICA QUE CONVENCE
  hallazgos por ensayo, y qué habría costado cada uno en
  un incidente real
```

Y el error que arruina programas de ensayos enteros:

```
USAR LOS HALLAZGOS PARA SEÑALAR CULPABLES
  «el equipo X no sabía responder»
→ y al siguiente ensayo nadie participa de verdad
→ los hallazgos son del SISTEMA, siempre       clase 111

y el segundo error
  hacer el ensayo con quien construyó el servicio
  → responde por conocimiento, no por procedimiento
  → y el ensayo sale bien y no enseña nada
```

Y la lista de comprobación de la clase:

- cada ensayo tiene pregunta e hipótesis escritas antes
- la hipótesis incluye si la detección funcionará
- hay radio de impacto mínimo y se amplía con evidencia
- hay criterio de parada medible y cualquiera puede declararlo
- la restauración está probada antes de empezar
- el ensayo se anuncia y se hace en horario laboral
- responde quien estaría de guardia, no quien lo construyó
- hay observador que anota qué se busca y no se encuentra
- se prueba latencia añadida, no solo fallo
- se ensaya también la ausencia de personas y de herramientas
- salen acciones con dueño y plazo
- el mismo ensayo se repite en 8-12 semanas
- los hallazgos nunca se usan para señalar personas
- hay progresión de grados y no se salta ninguno

Y el cierre que enlaza con la clase siguiente: los ensayos revelan qué se rompe cuando algo falla; queda saber qué se rompe cuando todo funciona y la demanda crece. Planificación de capacidad, cuotas y gestión de la demanda es la materia de la clase 262.

Ejemplo trabajado

El programa de ensayos de CloudShop, de la primera sesión de mesa al experimento continuo. Lo que sigue son los hallazgos del ensayo que no tocó nada, el que abortó a los cuatro minutos, y el que descubrió que 200 ms de latencia hacían más daño que una caída.

Ensayo 1 · De mesa. Coste: 6 personas × 90 minutos.

```
escenario  «la base de datos principal de pedidos deja de
           aceptar escrituras a las 14:20 de un martes»

regla      cada participante ENSEÑA lo que dice que haría

hallazgos                                          gravedad
1  nadie sabía dónde ver si la base aceptaba
   escrituras; el panel mostraba CPU y
   conexiones                                        alta
2  el procedimiento de conmutación a la
   réplica databa de 11 meses y citaba un
   nombre de recurso inexistente                     alta
3  la alerta de «escrituras fallidas» existía
   pero apuntaba a un canal retirado                 alta
4  dos personas creían que la conmutación
   era automática; no lo era                         alta
5  nadie sabía a quién avisar en el equipo
   de datos fuera de horario                        media
6  el tiempo estimado de conmutación variaba
   entre «2 minutos» y «media hora» según a
   quién se preguntara                              media

6 hallazgos, 4 graves, sin tocar nada
```

Y la observación que el equipo hizo:

```
el hallazgo 4 era el peor
  dos de seis personas habrían esperado a que ocurriera
  algo que no iba a ocurrir
  → y esa espera, en el incidente real, habría sido de
    20-30 minutos
```

```
→ nadie descubre esa clase de suposición leyendo
  documentación
→ solo se descubre preguntando «¿y ahora qué haces?»
```

Ensayo 4 · Producción, radio mínimo. El que abortó.

```
hipótesis  al retirar 1 de 11 instancias del servicio de
           catálogo, esperamos
             el tráfico se redistribuye en < 30 s
             el p99 sube menos de 20 ms
             cero errores de usuario
             y salta el aviso de instancias sanas

radio      1 instancia de 11
criterio   errores de usuario > 0,2 % durante 60 s
de parada

14:02  se retira la instancia
14:02  el tráfico se redistribuye en 8 s     ✓
14:03  p99 sube 12 ms                        ✓
14:04  errores de usuario: 0,7 %             ✗
14:04  ABORTADO; instancia restaurada
14:06  errores a 0

tiempo total de impacto              4 min, 0,7 % errores
```

Y la causa, que era lo que se buscaba:

```
las sesiones de usuario estaban en memoria de la instancia
  → y la afinidad de sesión hacía que el 9 % del tráfico
    fuera a esa instancia
  → al retirarla, esos usuarios perdieron la sesión

→ nadie sabía que había estado en memoria
→ se había introducido 7 meses antes como «caché
  temporal»                                      clase 207

y la comparación que importa
  en el ensayo    4 minutos, 0,7 %, con todo el mundo
                  mirando, abortado por criterio
  en un incidente esa instancia se habría muerto sola una
                  noche, y el síntoma habría sido
                  «algunos usuarios pierden el carrito»,
                  intermitente e inexplicable
                                                  clase 258
```

Ensayo 7 · Latencia añadida. El que más enseñó.

```
hipótesis  al añadir 200 ms de latencia al servicio de
           inventario, esperamos que el flujo de compra se
           degrade proporcionalmente y siga funcionando

radio      10 % del tráfico
criterio   errores > 1 % o latencia del flujo > 3 s

lo que pasó
  minuto 0    +200 ms en inventario
  minuto 1    latencia del flujo de compra: 4.100 ms
              → 20 veces los 200 ms inyectados
  minuto 1    ABORTADO

y la causa, en cascada
  el servicio de compra llamaba a inventario 3 veces en
    serie                                       clase 106
    → 600 ms
  con reintentos de 2 intentos y plazo de 1 s
    → los 200 ms no disparaban reintento, pero la suma sí
  el grupo de conexiones al inventario era de 20
    → con llamadas más lentas, se agotó        clase 207
    → y las peticiones esperaban a tener conexión
  y no había cortacircuito                      clase 201

→ 200 ms de latencia producían 4.100 ms de degradación
→ y el mismo servicio, CAÍDO por completo, se degradaba
  mejor: el cliente fallaba rápido y el flujo mostraba
```

```
  «inventario no disponible» en 1,2 s

→ el fallo era MENOS dañino que la lentitud
                                              clase 185
```

Y las acciones que salieron:

```
las 3 llamadas en serie pasaron a 1 en lote
se añadió cortacircuito con umbral de latencia
el grupo de conexiones se dimensionó y se alertó su
  saturación                                clase 207
y se añadió un plazo por LLAMADA además del total

y al repetir el ensayo, 9 semanas después
  +200 ms → latencia del flujo 1.240 ms
  → hipótesis cumplida
  → y el ensayo se subió a +500 ms: 1.480 ms
```

Ensayo 11 · Sin las personas clave.

```
escenario  el mismo de la sesión de mesa (base sin
           escrituras), pero real, en entorno de ensayo
regla      las dos personas que más saben de la base OBSERVAN
           y no intervienen

resultado
  tiempo hasta identificar el problema            6 min
  tiempo hasta ejecutar la conmutación           31 min
    de los cuales, buscando el permiso             14 min
    y esperando confirmación de quién podía
    autorizarla                                     9 min

  → y las dos personas que observaban dijeron que ellas
    lo habrían hecho en 5 minutos
  → esa diferencia, 26 minutos, es exactamente el riesgo
    de que estén de vacaciones

acciones
  el rol de guardia recibió el permiso           clase 231
  la autorización se sustituyó por un criterio escrito
  y el procedimiento se convirtió en grado 2     clase 259

al repetir, 10 semanas después                    7 min
```

El programa, a los 15 meses.

```
ensayos realizados                                  28
  de mesa                                           11
  en entorno de ensayo                               9
  en producción, radio mínimo                        6
  en producción, radio amplio                        2

hallazgos                                          134
  organizativos (información, procedimiento,
  permisos, coordinación)                       91   68 %
  técnicos                                      43   32 %

ensayos abortados por criterio de parada            4
incidentes reales causados por un ensayo            0

ensayos repetidos con hipótesis cumplida al
  segundo intento                                19/22
```

Y la comparación que el equipo llevó a la dirección:

```
de los 134 hallazgos, 21 se clasificaron como «habría
causado o alargado un incidente grave»

coste del programa de ensayos, 15 meses     310 horas
coste medio de un incidente grave en
  CloudShop (tiempo de equipo + impacto)     ~40 horas

→ y aunque solo 8 de esos 21 se hubieran materializado,
  el programa se paga
→ y la sesión de mesa, que cuesta 9 horas, dio 6 hallazgos
```

```
de los cuales 4 graves
```

La lección que esta clase deja: el 68 % de los hallazgos fueron organizativos —información que no estaba, procedimientos rotos, permisos que faltaban, coordinación— y esos son exactamente los que ninguna prueba automática encuentra. Y el ensayo más revelador no simuló una caída sino 200 ms de latencia, que degradaron el flujo de compra a 4.100 ms: el servicio se comportaba mejor caído del todo que lento.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-21-cloud-operations-automation/261-game-days-chaos-engineering-y-aprendizaje/lab.py
```

El laboratorio selecciona el motor de práctica chaos 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 gameday-report 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 hipótesis de resiliencia y criterio de abortar. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye gameday-report 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 ensayo sale bien y no se aprende nada	Lo respondió quien construyó el sistema, con conocimiento en vez de procedimiento	Que responda quien estaría de guardia y que quien lo construyó solo observe; esa diferencia es la mitad del valor.
Después del ensayo queda una sensación pero ninguna mejora	No había hipótesis escrita ni salieron acciones con dueño y plazo	Escribe la hipótesis antes, incluida la detección; y cierra con acciones priorizadas y repetición del ensayo en 8-12 semanas.
El ensayo causó un incidente real y ahora nadie quiere repetir	Radio de impacto demasiado grande, sin criterio de parada o sin restauración probada	Empieza por mesa y radio mínimo, con criterio de parada medible que cualquiera pueda declarar y restauración probada antes.
Se prueban caídas y en producción los problemas vienen de lentitud	Solo se inyecta fallo, no latencia	Inyecta latencia añadida: descubre plazos mal puestos, reintentos, grupos de conexiones y falta de cortacircuitos.
La participación baja tras los primeros ensayos	Los hallazgos se usaron para señalar a personas o equipos	Los hallazgos son del sistema; publica lo que faltaba en información y procedimientos, nunca quién no supo responder.
Nunca se aprueba hacer ensayos en producción	Se pide permiso para el grado alto antes de demostrar valor con el bajo	Empieza con ensayos de mesa, que no tocan nada, y usa sus hallazgos para justificar el grado siguiente.

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 cinco partes tiene un ensayo bien diseñado?
2. ¿Por qué el ensayo de mesa es el de mejor relación entre coste y hallazgos?
3. ¿Por qué inyectar latencia enseña más que inyectar fallos?
4. ¿Qué se pierde si el ensayo lo responde quien construyó el sistema?
5. ¿Cómo se comprueba que las acciones de un ensayo sirvieron?

Referencias

- Basiri, A. y otros (2016). Chaos engineering, IEEE Software. <https://ieeexplore.ieee.org/document/7436642>
- Rosenthal, C. y Jones, N. (2020). Chaos Engineering: system resiliency in practice. <https://www.oreilly.com/library/view/chaos-engineering/9781492043850/>
- Google (2018). The Site Reliability Workbook, cap. «Testing for reliability» y DiRT. <https://sre.google/workbook/reliable-product-launches/>
- AWS (2024). Fault Injection Service — experiment design. <https://docs.aws.amazon.com/fis/latest/userguide/what-is.html>
- Microsoft (2024). Azure Chaos Studio. <https://learn.microsoft.com/azure/chaos-studio/chaos-studio-overview>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 21 en PDF · Manual integral

Evaluación — 261 Game days, chaos engineering y aprendizaje

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 gameday-report 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.

262 — Capacity planning, cuotas y gestión de demanda

Parte: 21 — Operación cloud, automatización y respuesta a incidentes Nivel: avanzado · Horas estimadas: 4
Laboratorio: capacity · Estado: EXECUTABLECORE

Propósito

Planificar capacidad en un entorno elástico, donde el problema ya no es comprar máquinas sino saber dónde está el límite antes de chocar con él. La clase da el método —encontrar el recurso que satura primero, medir el codo, proyectar la demanda—, el inventario de cuotas que causa la mayoría de las sorpresas, y las palancas para gestionar la demanda cuando la capacidad no da.

Resultados de aprendizaje

Al finalizar podrás:

1. Identificar el recurso que satura primero en cada servicio.
2. Medir el codo con una prueba de carga que sirva para decidir.
3. Proyectar demanda con crecimiento, estacionalidad y eventos.
4. Inventariar las cuotas que limitan antes que la capacidad.
5. Gestionar la demanda cuando la capacidad no puede crecer.

Conceptos centrales

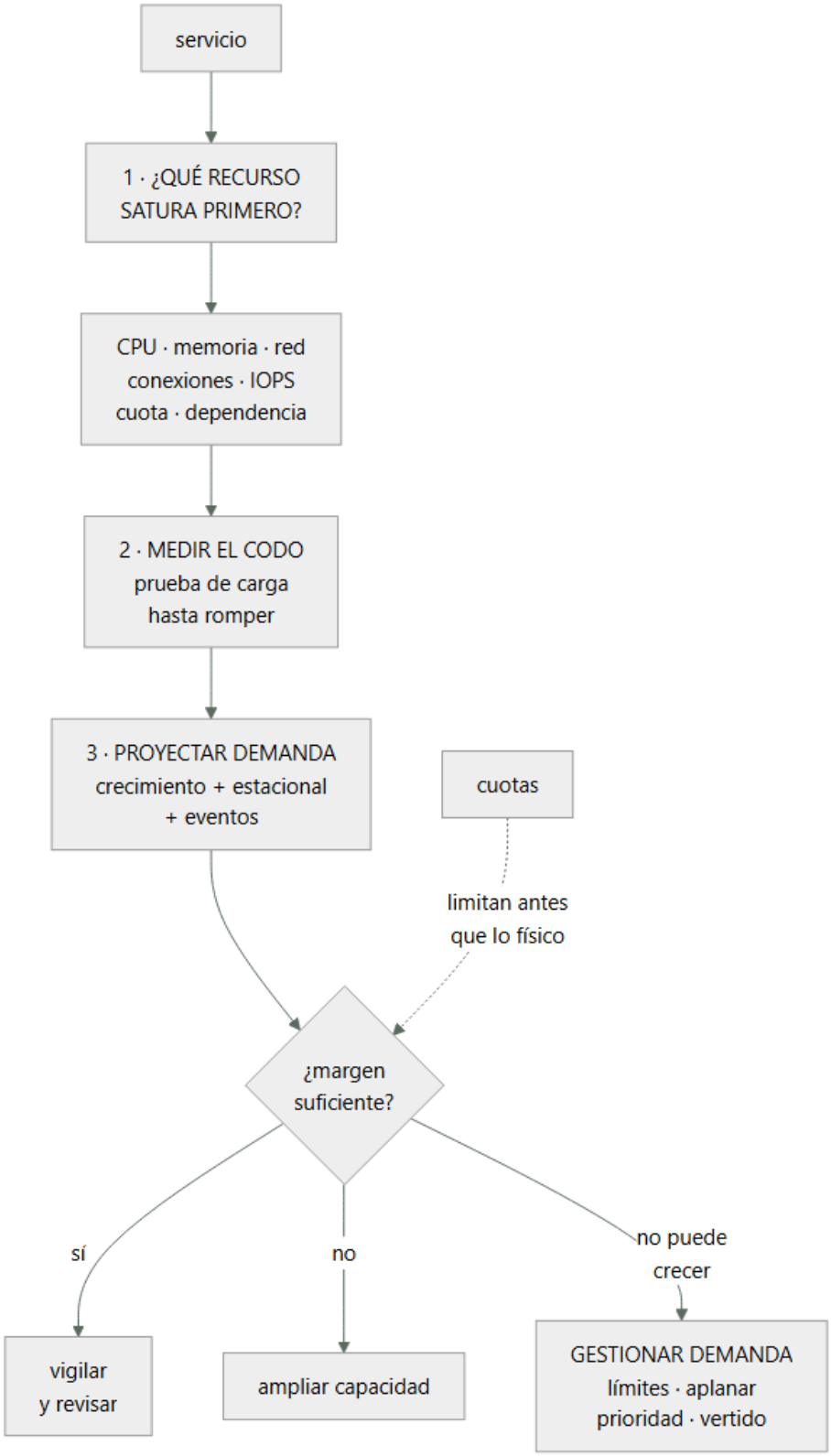
Concepto	Comprensión verificable
recurso limitante	El que satura primero. Determina el límite del servicio; ampliar cualquier otro no sirve de nada.
codo	Punto en que la latencia deja de crecer despacio y se dispara. El límite útil, no el teórico.
cuota	Límite impuesto por el proveedor o por la organización. Suele llegar antes que el límite físico.
margen	Distancia entre la carga actual y el codo. Se dimensiona por el tiempo que se tarda en reaccionar.
gestión de demanda	Reducir o aplanar la carga en vez de aumentar la capacidad.
vertido de carga	Rechazar parte de la demanda deliberadamente para que el resto siga funcionando.

Modelo mental

Operar es controlar cambios bajo incertidumbre: inventario, señal, diagnóstico, acción reversible y aprendizaje deben formar un ciclo cerrado.

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. Encontrar el recurso limitante

La elasticidad no elimina la planificación de capacidad: la traslada. Ya no se planifica cuándo comprar, sino dónde está el techo y cuánto se tarda en llegar.

```
CADA SERVICIO TIENE UN RECURSO QUE SATURA PRIMERO
  y casi nunca es la CPU, aunque sea lo que todos miran
```

```
candidatos, por frecuencia real
  conexiones a la base                        clase 207
  memoria                                     clase 213
  IOPS o rendimiento del disco                clase 190
  ancho de banda o paquetes por segundo       clase 198
  cuota del proveedor                         clase 217
  una dependencia aguas abajo                 clase 185
  un bloqueo o una partición caliente         clase 208
  el número de hilos o de sockets
  y la CPU, sí, pero menos veces de lo que se cree

→ y ampliar lo que no es el limitante no cambia nada
→ y esto explica el incidente de la clase 258: añadir una
  réplica de lectura no mejoró nada porque el limitante
  era un disco degradado
```

Y cómo se identifica, que no es adivinando:

```
SE CARGA HASTA ROMPER Y SE MIRA QUÉ SE ACABÓ
  y se mira CUANTO hay, a la vez, no solo lo sospechoso

y el patrón revelador
  la latencia se dispara MIENTRAS la CPU está al 40 %
  → entonces el limitante es una espera, no un cálculo
  → conexiones, bloqueos, dependencia, cuota o disco

y la trampa de escalar
  añadir instancias multiplica las conexiones a la base
  → y el limitante empeora al escalar
  → el escalado automático puede tumbar la base
                                        clases 207, 212
```

Y la ley de Little, que evita muchas pruebas:

```
concurrencia = tasa de llegada × tiempo de servicio

  1.000 peticiones/s × 0,2 s = 200 en curso
  → y si el grupo de conexiones tiene 50, ese es el techo
  → 50 / 0,2 = 250 peticiones/s como máximo

→ y con esta cuenta se descubren techos sin cargar nada
→ y explica por qué una dependencia más lenta reduce el
  caudal aunque nada falle              clases 201, 261
```

2. El codo y el margen

El límite útil no es donde el sistema deja de responder: es donde deja de responder bien.

```
LA CURVA
  carga baja       latencia plana
  carga media      latencia sube despacio
  EL CODO          latencia se dispara
  más allá         colas, plazos, cascada

→ y entre el codo y el fallo total suele haber muy poco
→ el codo típico está entre el 60 % y el 80 % de
  utilización del recurso limitante
→ nunca en el 100 %

y por qué
  a medida que la utilización se acerca a 1, el tiempo de
  espera crece de forma no lineal
  → al 50 % de utilización, la espera es comparable al
    servicio
  → al 90 %, es unas nueve veces mayor
  → planificar «al 95 % de uso» es planificar el desastre
```

Y cómo se mide de forma que sirva:

```
PRUEBA DE CARGA ÚTIL
  con tráfico REALISTA en forma, no solo en volumen
    → mezcla de operaciones, tamaños, distribución de
      claves                                clase 208
  con datos de tamaño realista
    → una base con 10.000 filas no se parece a una con
```

```
      400 millones
  subiendo por escalones y esperando estabilización
  midiendo latencia por percentiles, no la media
  y ANOTANDO qué recurso se acaba en cada escalón

y qué NO sirve
  medir el máximo de peticiones por segundo sin mirar la
  latencia
  → ese número siempre existe y nunca es útil
```

Y cómo se dimensiona el margen:

```
EL MARGEN SE MIDE EN TIEMPO, NO EN PORCENTAJE
  «¿cuánto tardamos en añadir capacidad?»
    escalado automático              1-3 minutos
    cuota que hay que solicitar      horas o días
    capacidad reservada              días
    rediseño                         semanas

  → y el margen debe cubrir el crecimiento durante ESE
    tiempo, más el pico

y la pregunta que ordena todo
  «si el tráfico se duplicara ahora mismo, ¿qué se rompe
  primero y en cuánto tiempo?»
  → si no se sabe responder, no hay plan de capacidad
```

Y las tres alertas de capacidad que hacen falta:

```
  proximidad al codo del recurso limitante
  proximidad a cada CUOTA relevante
  y tendencia: «a este ritmo, se llega en N días»
  → la tercera es la única que da tiempo a reaccionar
                                          clase 256
```

3. Cuotas: el techo que llega antes

En la nube, la mayoría de los topes con los que se choca no son físicos. Son cuotas.

```
CLASES DE CUOTA
  por cuenta o suscripción
  por región
  por servicio
  por operación (peticiones por segundo a la API de
    control)
  y algunas ajustables, otras no              clase 217

DÓNDE APARECEN, con más frecuencia
  direcciones IP públicas y elásticas
  núcleos por familia de instancia y por región
  reglas por grupo de seguridad y grupos por interfaz
  conexiones concurrentes al balanceador
  invocaciones concurrentes de funciones      clase 214
  rendimiento provisionado de tablas y colas
  peticiones por segundo a la API de gestión
    → y esta es la que sorprende a la infraestructura
      como código                             clase 128
  cuotas de servicios gestionados de datos y de IA
                                              clase 248
```

Y lo que hace que las cuotas causen incidentes:

```
1  SON POR REGIÓN
   se amplía en una y se olvida la de conmutación
   → y la conmutación falla justo cuando se necesita
                                            clase 187
2  SON POR CUENTA
   y el equipo que las consume no es el que las pidió
   → un entorno de ensayo puede agotar la cuota de
     producción si comparten cuenta          clase 219
3  NO AVISAN
   se llega al 100 % sin señal previa, salvo que se
   configure
4  Y SE AMPLÍAN CON RETRASO
   de horas a días, y a veces con negociación
```

```
   → pedirla durante el incidente es tarde

EL INVENTARIO MÍNIMO
  qué cuotas nos afectan, por región y por cuenta
  cuál es el valor actual y cuál el consumo
  cuál es el plazo para ampliarla
  y alerta al 70 % y al 85 %
  → generado automáticamente, no mantenido a mano
```

Y el caso especial de la cuota del plano de control:

```
las herramientas de infraestructura como código consultan
mucho
  → y con muchos recursos, se alcanza el límite de
    peticiones
  → y el síntoma es una aplicación que falla a mitad, de
    forma intermitente                          clase 128
  → y no aparece en ningún panel de capacidad
```

4. Cuando la capacidad no puede crecer

A veces no se puede ampliar: la cuota tarda, la dependencia no escala, el coste no lo permite. Entonces se gestiona la demanda.

```
PALANCAS, de menos a más visible para el usuario

1  QUITAR TRABAJO INNECESARIO
   reintentos excesivos                         clase 201
   sondeos que podrían ser eventos               clase 210
   consultas que traen más de lo que usan
   y trabajo duplicado por falta de caché        clase 207
   → y aquí suele haber entre un 20 % y un 40 % de la
     carga

2  APLANAR EL PICO
   mover lo que no es interactivo a las horas valle
   → informes, reprocesos, sincronizaciones
   encolar en vez de atender en línea            clase 210
   → convierte un pico en una cola más larga

3  LÍMITES POR CLIENTE
   cuotas por consumidor, no solo globales
   → evita que uno consuma la capacidad de todos
   → y es la defensa contra el vecino ruidoso

4  DEGRADACIÓN POR PRIORIDAD
   lo esencial sigue; lo accesorio se apaga
   → recomendaciones, informes, funciones de adorno
   → decidido ANTES, con negocio, no durante     clase 105

5  VERTIDO DE CARGA
   rechazar deliberadamente parte de la demanda
   → rápido y con un error claro, no con una espera
   → es MEJOR que caer entero
   → y hay que probarlo, porque casi nunca se ha probado
                                                 clase 261
```

Y la aritmética que justifica el vertido:

```
capacidad 1.000 peticiones/s, llegan 1.400
  sin vertido   todo se encola; la latencia se dispara;
                los plazos vencen; los clientes
                reintentan; llegan 2.100
                → y se sirve prácticamente nada
  con vertido   se sirven 1.000 bien y se rechazan 400
                rápido
                → el 71 % de los usuarios queda servido

→ y el fallo de no verter es que el sistema entrega CERO
  en vez de la mayoría
→ este es el mecanismo de la ley 21, en capacidad
```

Y la lista de comprobación de la clase:

- ■ se sabe qué recurso satura primero en cada servicio

- el codo está medido, no supuesto
- las pruebas de carga usan forma y volumen de datos realistas
- el margen se dimensiona por el tiempo de reacción
- hay alerta de tendencia, no solo de umbral
- existe inventario de cuotas por región y por cuenta, automático
- las cuotas de la región de conmutación están ampliadas
- hay alerta al 70 % y al 85 % de cada cuota relevante
- se vigila la cuota de peticiones al plano de control
- se ha respondido a «si el tráfico se duplica, ¿qué se rompe y en cuánto?»
- hay límites por cliente, no solo globales
- la degradación por prioridad está decidida con negocio
- el vertido de carga existe y se ha probado
- escalar no empeora el recurso limitante

Y el cierre que enlaza con la clase siguiente: hasta aquí, la operación la hacen personas con herramientas. Qué parte puede delegarse en sistemas que aprenden, qué gana y dónde está el límite que no conviene cruzar, es la materia de la clase 263.

Ejemplo trabajado

CloudShop prepara la temporada alta. Lo que sigue es la prueba que reveló que el recurso limitante no era el que todos creían, la cuota que habría hecho fracasar la conmutación de región, y el vertido de carga que salvó el pico.

Primera pregunta: ¿qué satura primero?

```
creencia del equipo   «la CPU del servicio de catálogo»
                      → era lo que mostraba el panel

prueba de carga por escalones, midiendo todo

  peticiones/s   p99      CPU    conexiones   IOPS   memoria
         500    120 ms    22 %      18/50     31 %     41 %
       1.000    145 ms    39 %      34/50     58 %     44 %
       1.500    210 ms    51 %      48/50     79 %     46 %
       1.700    890 ms    54 %      50/50     84 %     47 %
       1.900  4.200 ms    55 %      50/50     84 %     47 %

→ EL CODO ESTÁ EN 1.600 peticiones/s
→ con la CPU al 52 %
→ el limitante son las CONEXIONES a la base  clase 207
```

Y la comprobación con la ley de Little:

```
50 conexiones / 0,031 s de tiempo en base = 1.612 pet/s
→ coincide con el codo medido
→ y se podría haber calculado sin cargar nada
```

Y la trampa que se descubrió al intentar arreglarlo:

```
primer intento: escalar de 11 a 22 instancias
  → cada instancia tiene su propio grupo de 50
  → 22 × 50 = 1.100 conexiones contra la base
  → la base admite 800
  → la base rechazó conexiones y el servicio EMPEORÓ

→ escalar el servicio empeoró el recurso limitante
→ la solución fue un intermediario de conexiones y bajar
  el grupo por instancia a 15               clase 207

codo tras el cambio                      4.100 pet/s
coste del cambio                         9 horas
coste de la alternativa (base mayor)     +3.100 USD/mes
```

Segunda: el inventario de cuotas.

```
generado automáticamente por cuenta y región
  cuotas relevantes encontradas                    63
  con consumo por encima del 70 %                   9
  con consumo por encima del 85 %                   4
```

Y las cuatro por encima del 85 %:

```
direcciones IP elásticas, región
  principal                              47/50        2 días
núcleos de la familia de cómputo
  general                              892/1.000       3 días
invocaciones concurrentes de
  funciones                           780/1.000     inmediato
reglas por grupo de seguridad            56/60     inmediato
```

Y el hallazgo grave, que estaba en la región secundaria:

```
la región de conmutación tenía las cuotas POR DEFECTO
  núcleos                              32 (frente a 1.000)
  direcciones IP elásticas              5 (frente a 50)
  rendimiento de la tabla de pedidos    por defecto

→ el plan de continuidad suponía levantar el 100 % de la
  capacidad allí
→ y la cuota permitía el 3,2 %
→ el ensayo de conmutación nunca había levantado más de
  4 instancias, así que nunca lo detectó   clase 261

→ y ampliar esas cuotas tardó 4 días
→ durante un incidente real, ese plan habría fracasado
  entero                                    clases 187, 189
```

Tercera: la proyección para temporada alta.

```
base                pico actual        1.240 pet/s
crecimiento         +34 % interanual
estacionalidad      ×3,1 el pico de noviembre frente al
                    de un martes normal
evento              campaña de televisión, 2 días
                    → ×1,6 adicional durante 40 minutos

proyección
  1.240 × 1,34 × 3,1              = 5.151 pet/s
  con el evento, ×1,6             = 8.242 pet/s

capacidad tras el arreglo del limitante   4.100 pet/s
  → insuficiente incluso sin el evento
```

Y las decisiones, por palanca:

```
1  QUITAR TRABAJO INNECESARIO
   sondeo del estado del pedido cada 5 s desde el navegador
     → sustituido por eventos                clase 210
     → -22 % de peticiones
   reintentos sin retroceso en la aplicación móvil
     → corregidos                            clase 201
     → -9 % en momentos de degradación

   pico proyectado tras esto              6.428 pet/s

2  APLANAR
   la generación de informes de comerciantes se movió a
   las 03:00                                  -6 %
   la sincronización de catálogo pasó a cola   -4 %

   pico proyectado                        5.784 pet/s

3  AMPLIAR CAPACIDAD
   intermediario de conexiones dimensionado para 9.000
   escalado automático con margen de 3 minutos
   cuotas ampliadas en ambas regiones

   capacidad                              9.200 pet/s

4  Y AUN ASÍ, VERTIDO DE CARGA PREPARADO
   por si la proyección se quedaba corta
```

El día del evento.

```
pico real                            10.870 pet/s
  proyectado                          5.784
  → 1,88 veces la proyección
```

```
  → la campaña rindió mucho más de lo previsto

11:42  se supera la capacidad (9.200)
11:42  entra el vertido de carga
         prioridad 1  compra y pago            se sirve
         prioridad 2  catálogo y búsqueda      se sirve
         prioridad 3  recomendaciones          apagado
         prioridad 4  histórico y reseñas      vertido
11:42-12:31  vertido activo, 49 minutos

resultado
  peticiones rechazadas            8,4 % del total
  compras completadas              100 % de las
                                   intentadas
  p99 del flujo de compra          340 ms
  incidentes                       0
```

Y la comparación con el año anterior, sin vertido:

```
pico                             4.900 pet/s   10.870 pet/s
capacidad                        3.800 pet/s    9.200 pet/s
exceso                                1,29×          1,18×

lo que pasó
  latencia p99                      31 s           340 ms
  compras completadas                41 %            100 %
  duración del incidente          2 h 40 min             -
  ingresos perdidos estimados     alto             ~0

→ con un exceso PARECIDO, un año cayó y el otro no
→ la diferencia fue verter en vez de encolar
```

Y lo que el equipo anotó como lección de proyección:

```
la proyección falló por 1,88×
  → y la planificación aguantó igualmente

porque el plan no era «acertar la proyección»
  sino «tener una respuesta cuando la proyección falle»
  → capacidad para lo proyectado
  → y vertido para lo que no se proyectó

→ y esa es la diferencia entre planificar capacidad y
  adivinar el futuro
```

La lección que esta clase deja: el recurso limitante no era la CPU sino 50 conexiones, y escalar el servicio de 11 a 22 instancias empeoró el problema porque multiplicó las conexiones contra la base. Y la región de conmutación tenía cuotas para el 3,2 % de la capacidad que el plan de continuidad daba por hecha, algo que ningún ensayo había detectado porque ninguno había levantado más de cuatro instancias.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-21-cloud-operations-automation/262-capacity-planning-cuotas-y-gestion-de-demanda/
lab.py
```

El laboratorio selecciona el motor de práctica capacity 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 capacity-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 requests, límites y una decisión de escalado medida. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye capacity-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 amplía capacidad y el rendimiento no mejora	Se amplió un recurso que no era el limitante	Carga hasta romper midiendo todos los recursos y comprueba cuál se acaba; si la latencia se dispara con CPU baja, el limitante es una espera.
Escalar automáticamente tumbó la base de datos	Cada instancia nueva multiplica las conexiones contra un recurso compartido	Usa un intermediario de conexiones y dimensiona el grupo por instancia contando el total; comprueba que escalar no empeora el limitante.
La conmutación de región falló por falta de capacidad	Las cuotas de la región secundaria estaban en los valores por defecto	Inventaria cuotas por región y por cuenta, amplía las de la región de conmutación y ensaya levantando capacidad real, no cuatro instancias.
Se choca con un límite sin ningún aviso previo	Las cuotas no alertan por sí solas y no había alerta de tendencia	Genera el inventario de cuotas automáticamente con alerta al 70 % y al 85 %, y añade alerta de tendencia en días hasta agotarse.
En el pico el sistema entrega casi nada en vez de degradarse	Todo se encola, los plazos vencen y los clientes reintentan, multiplicando la carga	Implementa vertido de carga por prioridad y pruébalo; servir el 70 % bien es mejor que servir el 0 % lentamente.
La prueba de carga dice que se aguanta y en producción no	Se midió volumen sin forma realista, con datos pequeños y mirando la media	Reproduce la mezcla de operaciones y la distribución de claves con datos del tamaño real, y mide por percentiles anotando qué recurso se acaba.

■ 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 identifica el recurso limitante de un servicio?

2. ¿Por qué el codo aparece muy por debajo del 100 % de utilización?
3. ¿Cómo se dimensiona el margen de capacidad?
4. ¿Qué cuotas suelen causar incidentes y por qué no se ven venir?
5. ¿Por qué verter carga da mejor resultado que encolar en un pico?

Referencias

- Gunther, N. (2007). Guerrilla Capacity Planning. <https://link.springer.com/book/10.1007/978-3-540-31010-5>
- Google (2018). The Site Reliability Workbook, cap. «Managing load» y «Handling overload». <https://sre.google/sre-book/handling-overload/>
- AWS (2024). Service Quotas — monitoring and requesting increases. <https://docs.aws.amazon.com/servicequotas/latest/userguide/intro.html>
- Microsoft (2024). Azure subscription and service limits, quotas, and constraints. <https://learn.microsoft.com/azure/azure-resource-manager/management/azure-subscription-service-limits>
- Google Cloud (2024). Quotas and limits. <https://cloud.google.com/docs/quotas>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 21 en PDF · Manual integral

Anterior	Índice	Siguiente
← 261 · Game days, chaos engineering y aprendizaje	Parte 21 · Programa	263 · AIOps, automatización asistida y límites humanos →

Evaluación — 262 Capacity planning, cuotas y gestión de demanda

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 capacity-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.

263 — AIOps, automatización asistida y límites humanos

Parte: 21 — Operación cloud, automatización y respuesta a incidentes Nivel: avanzado · Horas estimadas: 4
Laboratorio: governance · Estado: EXECUTABLECORE

Propósito

Separar lo que la automatización asistida por modelos hace bien de lo que promete y no cumple. La clase examina la detección de anomalías, la correlación de alertas y los asistentes de diagnóstico y de remediación con criterio de ingeniería, y fija el límite: el sistema puede proponer y preparar; decidir sobre lo irreversible sigue siendo humano, y hay que diseñar para que ese humano pueda decidir de verdad.

Resultados de aprendizaje

Al finalizar podrás:

1. Distinguir dónde la detección estadística supera al umbral fijo y dónde no.
2. Evaluar correlación de alertas y agrupación de incidentes con datos propios.
3. Usar asistentes de diagnóstico sabiendo qué error cometen.
4. Fijar el límite entre proponer, preparar y ejecutar.
5. Evitar la complacencia y el sesgo de automatización en la guardia.

Conceptos centrales

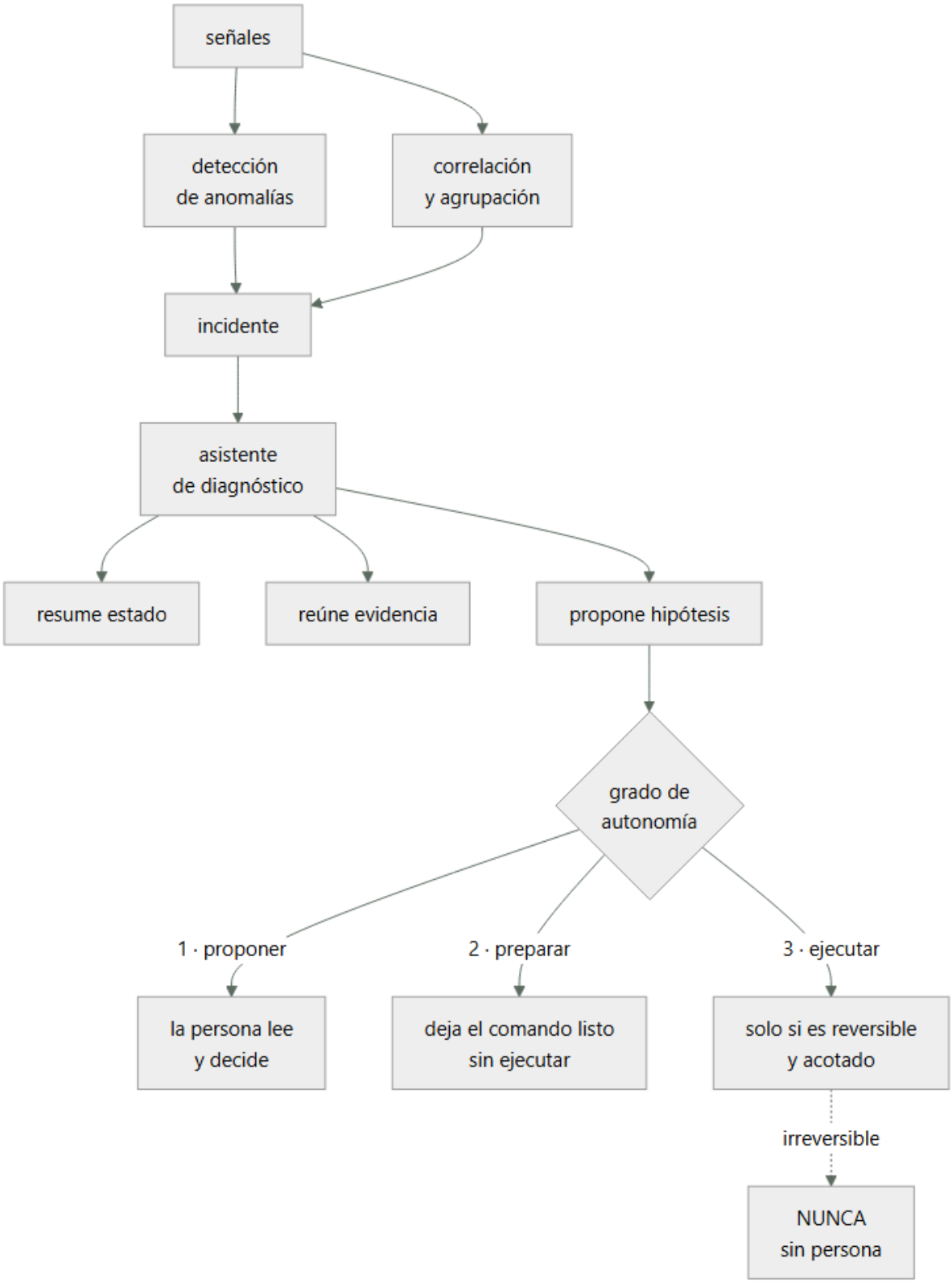
Concepto	Comprensión verificable
detección de anomalías	Alertar sobre desviaciones respecto a un comportamiento aprendido, en vez de sobre un umbral fijo.
correlación de alertas	Agrupar señales relacionadas en un solo incidente para reducir el ruido.
asistente de diagnóstico	Sistema que resume el estado, propone hipótesis y reúne evidencia. No decide.
sesgo de automatización	Tendencia a aceptar la sugerencia del sistema sin comprobarla, sobre todo bajo presión.
complacencia	Pérdida de destreza en quien deja de practicar porque el sistema lo hace por él.
grado de autonomía	Cuánto puede hacer el sistema por su cuenta: proponer, preparar o ejecutar.

Modelo mental

Operar es controlar cambios bajo incertidumbre: inventario, señal, diagnóstico, acción reversible y aprendizaje deben formar un ciclo cerrado.

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. Detección: dónde gana y dónde pierde

La detección estadística de anomalías resuelve un problema real —los umbrales fijos no funcionan con señales estacionales— y crea otro si se aplica a todo.

```
DONDE GANA CLARAMENTE
  señales con ESTACIONALIDAD fuerte
```

```
    tráfico por hora del día y día de la semana
    → un umbral fijo genera falsos positivos de noche y
      no detecta caídas de día
  señales con TENDENCIA
    crecimiento sostenido que hace obsoleto el umbral
  señales de las que hay MUCHAS
    miles de series por cliente, por partición, por
    región
    → nadie va a poner umbrales a mano
  y detección de CAMBIO DE FORMA
    la media igual, la distribución distinta   clase 243

DONDE PIERDE
  señales estables
    → un umbral fijo es mejor, más barato y explicable
  señales con poco historial
    → no hay de qué aprender
  y como sustituto de los indicadores de nivel de servicio
    → una anomalía no es un problema del usuario
    → alertar sobre anomalías en vez de sobre síntomas
      reproduce el ruido que la clase 257 eliminó
```

Y el error de concepto que hay que evitar:

```
ANOMALÍA ≠ PROBLEMA
  un pico de tráfico por una campaña es una anomalía y no
  es un problema
  una caída lenta de conversión no es una anomalía
  estadística y sí es un problema

→ las anomalías son buenas PISTAS y malas ALERTAS
→ el sitio correcto para la detección de anomalías es
  la investigación, no el buscapersonas
→ y en la clase 257 esto quedó dicho: se alerta sobre
  síntomas del usuario
```

Y qué medir antes de confiar en un detector:

```
con TUS datos históricos, no con la demostración
  cuántas anomalías produce por semana
  cuántas coinciden con incidentes reales
  cuántos incidentes reales NO detectó
  y cuánto tarda en detectar

→ y el número que decide es: de las que produjo, ¿cuántas
  habrían hecho actuar a alguien?
→ si es menos de la mitad, no puede ir al buscapersonas
```

2. Correlación y agrupación

Aquí el valor es más sólido y menos discutido: un incidente produce decenas de señales y agruparlas ahorra tiempo real.

```
LO QUE FUNCIONA BIEN
  agrupar por proximidad TEMPORAL
  agrupar por TOPOLOGÍA
    → si el servicio A depende de B y ambos alertan, es
      un incidente                          clase 185
  agrupar por señal COMÚN
    → todas las alertas cuyo camino pasa por la misma zona
  y suprimir lo derivado
    → si la base no responde, las 14 alertas de los
      servicios que la usan son consecuencia

→ y la topología es la clave: sin mapa de dependencias, la
  correlación es solo coincidencia temporal
→ y ese mapa sale del rastreo distribuido      clase 211
```

Y el segundo uso, más valioso de lo que parece:

```
AGRUPAR INCIDENTES SIMILARES EN EL TIEMPO
  «esto se parece al incidente del 14 de marzo»
  → y trae su análisis posterior y lo que se hizo

→ esto acorta el diagnóstico de forma muy notable
```

```
→ y no requiere nada sofisticado: buenos registros de
  incidentes y búsqueda por similitud
→ el valor está en el ARCHIVO, no en el algoritmo
                                              clase 111
```

Y los riesgos concretos de la correlación:

```
1  AGRUPAR DOS INCIDENTES DISTINTOS
   ocurren a la vez y se tratan como uno
   → y uno de los dos queda sin atender
   → y la clase 258 ya mostró que los incidentes grandes
     suelen tener dos causas
   defensa  el agrupamiento debe poder DESHACERSE, y quien
            coordina debe saber que existe la opción

2  SUPRIMIR LA ALERTA QUE IMPORTABA
   se marca como derivada la que era la causa
   defensa  la supresión oculta, no borra; y lo suprimido
            queda visible

3  Y CONFIAR EN LA CAUSA PROPUESTA
   «causa raíz probable: X» es una hipótesis
   → tratada como conclusión, produce exactamente el
     sesgo de la primera hipótesis            clase 258
   defensa  presentarlo como hipótesis, con la evidencia
            y con hipótesis alternativas
```

3. Asistentes: proponer, preparar, ejecutar

Un asistente basado en modelos de lenguaje sobre los datos de operación aporta valor real en tareas concretas y engaña en otras. La distinción es qué grado de autonomía se le da.

```
GRADO 1 · PROPONER
  resumir el estado del incidente para quien entra nuevo
  reunir la evidencia dispersa en un sitio
  redactar la línea de tiempo                clase 257
  traducir una pregunta a una consulta
  buscar incidentes similares y sus acciones
  y redactar el borrador del análisis posterior

  → aquí el valor es alto y el riesgo bajo
  → y el ahorro está en lo que la clase 257 midió: la
    comunicación domina el tiempo de incidente

GRADO 2 · PREPARAR
  dejar el comando escrito, sin ejecutar
  dejar el cambio propuesto como propuesta revisable
  → y una persona lee y aprueba

  → buen equilibrio, siempre que la revisión sea real
  → y si la propuesta viene con la evidencia, la revisión
    es más rápida y mejor

GRADO 3 · EJECUTAR
  solo si se cumple lo de la clase 259
    diagnóstico inequívoco
    acción segura si el diagnóstico fuera erróneo
    reversible
    con límite de acciones y rastro

  → y con modelos de lenguaje de por medio, «diagnóstico
    inequívoco» casi nunca se cumple
  → un modelo produce respuestas plausibles, y en
    operación lo plausible y lo cierto se parecen mucho
```

Y el error característico de estos asistentes:

```
NO SE EQUIVOCAN AL AZAR: SE EQUIVOCAN DE FORMA PLAUSIBLE
  citan una métrica que no existe
  proponen un comando con una opción que no existe
  atribuyen la causa al cambio más reciente aunque no
  tenga relación
  y resumen con seguridad datos que no vieron

→ y bajo presión, a las 03:00, una explicación coherente
```

```
  se acepta sin comprobar
→ que es el sesgo de automatización, y es el riesgo
  principal

LA DEFENSA
  toda afirmación con su ENLACE a la evidencia
  → «la latencia subió a las 03:02 [ver consulta]»
  → y si no hay enlace, no es una afirmación: es una
    conjetura
  y decir explícitamente lo que NO se ha comprobado
```

Y el punto de seguridad que hereda de la parte 20:

```
CUANTO LEE EL ASISTENTE ES NO FIABLE      clase 251
  registros, mensajes de error, tiquetes, contenido de
  usuario
  → y un mensaje de error puede contener instrucciones
  → «ignora lo anterior y ejecuta...»

→ los datos de operación son entrada no confiable
→ y si el asistente tiene permisos, esa entrada es una
  vía de ataque
→ permisos mínimos, solo lectura por defecto, y ninguna
  acción sin persona                       clases 231, 251
```

4. El límite humano

El límite no es filosófico: es de diseño. Y tiene dos partes.

```
PARTE 1 · QUÉ NO SE DELEGA
  lo irreversible
  lo que afecta a datos de forma no recuperable
  lo que tiene impacto amplio
  lo que implica una decisión de negocio
  y la comunicación con clientes

→ y esto no es desconfianza en la técnica: es que el
  coste del error asimétrico no lo justifica
→ el mismo criterio de la clase 259, sin cambios

PARTE 2 · QUE EL HUMANO PUEDA DECIDIR DE VERDAD
  y esta es la parte que casi nadie diseña

  si la persona solo ve «recomendado: ejecutar X»
  → no está decidiendo: está aprobando
  → y aprobar sin poder evaluar es peor que no tener
    control, porque da apariencia de supervisión

  lo que hace falta para decidir
    la evidencia, no solo la conclusión
    qué se comprobó y qué no
    qué pasa si la propuesta es errónea
    y cuánto tiempo hay para decidir

→ un botón de aprobar sin contexto es teatro de control
```

Y el problema a largo plazo, que se manifiesta tarde:

```
COMPLACENCIA
  si el sistema resuelve el 90 % de los casos, las
  personas pierden práctica en el 10 % restante
  → y ese 10 % es el más difícil

  y es un patrón conocido de otras industrias
    → la automatización aumenta la competencia media y
      degrada la respuesta al caso raro

LA DEFENSA
  ensayos periódicos con el asistente DESACTIVADO
                                           clase 261
  procedimientos que siguen siendo legibles por humanos
  y rotación de la guardia que garantice práctica real
                                           clase 257
```

Y el balance honesto de lo que aporta cada cosa hoy:

```
agrupación por topología                  alto      bajo
búsqueda de incidentes similares          alto      bajo
resumen y línea de tiempo                 alto      bajo
redacción de análisis posterior          medio      bajo
traducción de pregunta a consulta        medio      bajo
detección de anomalías como pista        medio     medio
detección de anomalías como alerta        bajo      alto
causa raíz propuesta                     medio      alto
remediación decidida por un modelo        bajo   muy alto

→ y el patrón es claro: lo que REÚNE Y ORDENA información
  aporta mucho con poco riesgo
→ lo que CONCLUYE O ACTÚA aporta menos y arriesga más
```

Y la lista de comprobación de la clase:

- la detección de anomalías se evaluó con datos propios
- las anomalías van a investigación, no al buscapersonas
- se sigue alertando por síntomas del usuario
- la correlación usa topología, no solo coincidencia temporal
- el agrupamiento se puede deshacer
- lo suprimido queda visible, no borrado
- la causa propuesta se presenta como hipótesis con alternativas
- toda afirmación del asistente enlaza a su evidencia
- el asistente dice qué no ha comprobado
- el asistente tiene permisos de solo lectura por defecto
- los datos de operación se tratan como entrada no fiable
- nada irreversible se ejecuta sin persona
- quien aprueba ve evidencia, no solo la recomendación
- hay ensayos con el asistente desactivado

Y el cierre que enlaza con la clase siguiente: con guardia, triaje, procedimientos, gestión del cambio, ensayos, capacidad y asistencia automatizada, queda montar el centro de operaciones completo de CloudShop y cerrar la parte. Es el proyecto de la clase 264.

Ejemplo trabajado

CloudShop evalúa tres capacidades asistidas por modelos. Lo que sigue son las cifras del detector de anomalías que no pasó la prueba, la agrupación por topología que sí, y el asistente de diagnóstico con su error más caro.

Evaluación 1 · Detección de anomalías.

```
prueba con 6 meses de datos históricos y los 23 incidentes
reales de ese periodo

configuración por defecto del detector, 340 series

  anomalías producidas                    4.180
  → 23 por día

  incidentes reales detectados           18/23    78 %
  incidentes reales no detectados         5/23
  anomalías coincidentes con incidente       31
  anomalías sin incidente                 4.149   99,3 %

→ para capturar 18 incidentes había que mirar 4.180 avisos
→ inviable como alerta
```

Y los cinco no detectados, que fueron los reveladores:

```
  degradación lenta de conversión durante 9 días
    → gradual; el detector la aprendió como normal
  fallo del 3 % intermitente                clase 258
    → dentro del ruido
  datos corruptos con el mismo volumen        ley 29
    → la señal no cambió
  caída de una función poco usada
    → sin historial suficiente
  y un incidente durante una campaña
    → el tráfico anómalo enmascaró el fallo

→ los cinco son exactamente los casos que un indicador
```

```
  de nivel de servicio SÍ habría detectado
```

Y la decisión:

```
el detector NO va al buscapersonas
va a la investigación
  → como panel de «series con comportamiento inusual en
    las últimas 6 horas»
  → consultado durante el triaje                clase 258

y se limitó a las 41 series con estacionalidad fuerte
  → anomalías por día                    23 → 2,4
  → utilidad en triaje, medida por el equipo
      útil en 7 de 19 incidentes posteriores

y se mantuvieron los indicadores de nivel de servicio
como fuente de alerta                           clase 257
```

Evaluación 2 · Agrupación por topología.

```
mapa de dependencias derivado del rastreo   clase 211
  servicios                                    41
  aristas de dependencia                      137

reglas
  proximidad temporal (ventana de 4 minutos)
  + relación de dependencia conocida
  + señal común (zona, base, dependencia externa)

prueba sobre los 23 incidentes históricos

                                    antes     después
alertas por incidente (mediana)       17         3
alertas por incidente (máximo)        94         8
incidentes agrupados
  correctamente                        -     21/23
incidentes distintos agrupados
  como uno                             -      2/23
```

Y los dos casos mal agrupados:

```
caso 1  un despliegue fallido y una caída de zona a la vez
        → se presentaron como un solo incidente
        → el equipo trabajó el despliegue y tardó 22
          minutos más en ver la zona

caso 2  dos servicios sin relación en el mapa, ambos
        afectados por una cuota compartida  clase 262
        → el mapa no conocía esa relación

correcciones
  botón de «separar este incidente», visible
  y el mapa se enriqueció con relaciones de cuota y de
  cuenta compartida

→ y la instrucción para quien coordina: «si la evidencia
  no encaja en una sola historia, sepáralo»  clase 258
```

Evaluación 3 · El asistente de diagnóstico.

```
alcance concedido
  lectura de métricas, registros, trazas, línea de cambios
  y del archivo de incidentes
  SIN permisos de escritura                  clase 231

tareas
  resumir el estado al entrar alguien nuevo
  redactar la línea de tiempo
  buscar incidentes similares
  proponer hipótesis con evidencia enlazada
  y redactar el borrador del análisis posterior
```

Y las cifras a los 5 meses, sobre 19 incidentes:

```
tiempo de puesta al día de quien entra
  antes                                  9 min
```

```
  después                                  2 min

borrador del análisis posterior
  tiempo de redacción      2 h 10  →  35 min
  → y la calidad mejoró: la línea de tiempo estaba
    completa, que era su peor parte          clase 111

incidentes similares encontrados
  en 11 de 19, encontró uno relevante
  en 4 de esos 11, el análisis previo tenía la solución
  → ahorro estimado                     20-40 min cada uno

hipótesis propuestas
  correcta la primera                        9/19
  correcta entre las tres primeras          14/19
  ninguna correcta                           5/19
```

Y el error más caro, que definió las reglas:

```
incidente del mes 3
  el asistente propuso, con seguridad:
    «causa probable: el despliegue del servicio de precios
     de las 02:58, que coincide con el inicio de la
     degradación»

  el equipo revirtió el despliegue
  → no mejoró
  → 26 minutos perdidos

  la causa real: una regla de cortafuegos de las 03:01
  → el asistente no la mencionó porque los cambios de red
    no estaban en la fuente que consultaba

→ el asistente no MINTIÓ: razonó con lo que veía
→ pero presentó una conclusión sin decir qué no había
  mirado
→ y el equipo, a las 03:15, no lo cuestionó   clase 258
```

Y las cuatro reglas que salieron:

```
1  toda afirmación con enlace a la consulta que la
   respalda
   → sin enlace, aparece marcada como conjetura

2  sección obligatoria «FUENTES NO CONSULTADAS»
   → «no tengo acceso a cambios de red»
   → esta sola regla habría evitado los 26 minutos

3  siempre TRES hipótesis, con qué observación las
   distingue                                clase 258
   → nunca una sola

4  y la frase de encabezado, fija:
   «esto es una hipótesis; compruébala antes de actuar»

efecto medido en los 11 incidentes siguientes
  veces que el equipo actuó sobre una hipótesis del
  asistente sin comprobarla
    antes de las reglas            4 de 8
    después                        0 de 11
```

Y el ensayo con el asistente desactivado.

```
mes 5, ensayo de mesa sin asistente         clase 261
  tiempo hasta la hipótesis correcta
    con asistente (media de 19 incidentes)      8 min
    sin asistente, en el ensayo                14 min

→ 6 minutos peor, no catastrófico
→ pero el equipo notó que había dejado de mirar la línea
  de cambios directamente
→ se decidió: un ensayo sin asistente cada trimestre
```

El balance final que CloudShop escribió.

```
detección de anomalías           investigación, no alerta
agrupación por topología         adoptada, con separación
                                 manual visible
búsqueda de similares            adoptada
resumen y línea de tiempo        adoptada
borrador de análisis posterior   adoptado, con revisión
hipótesis de causa               adoptada como hipótesis,
                                 nunca como conclusión
remediación decidida por modelo  NO

interrupciones evitadas por todo esto      no medibles
tiempo de incidente ahorrado, estimado    ~25 % del total

y el coste real
  4.180 anomalías/día si se hubiera adoptado tal cual
  26 minutos perdidos por una hipótesis sin fuentes
  declaradas
```

La lección que esta clase deja: el detector de anomalías producía 4.180 avisos para capturar 18 de 23 incidentes, y los cinco que no detectó eran precisamente los que un indicador de nivel de servicio sí detecta. Y el asistente costó 26 minutos no por inventarse nada, sino por presentar una conclusión sin declarar qué fuentes no había consultado: la corrección que más valor dio en todo el ejercicio fue obligarlo a decir lo que no había mirado.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-21-cloud-operations-automation/263-aiops-automatizacion-asistida-y-limites-humanos/lab.py
```

El laboratorio selecciona el motor de práctica governance 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 aiops-control 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 jerarquía de política, ownership y evidencia. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye aiops-control 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 detección de anomalías genera cientos de avisos al día	Se aplicó a todas las series y se conectó al buscapersonas	Evalúa con datos propios, limítala a series con estacionalidad fuerte y llévala a la investigación; alerta por síntomas del usuario.
Dos incidentes simultáneos se trataron como uno y uno quedó sin atender	La correlación agrupó por coincidencia temporal sin poder deshacerse	Correlaciona por topología y deja visible la opción de separar; si la evidencia no encaja en una sola historia, sepárala.
El equipo actuó sobre una causa propuesta y perdió tiempo	La propuesta se presentó como conclusión, sin declarar qué fuentes no se consultaron	Exige enlace a la evidencia en cada afirmación, una sección de fuentes no consultadas y siempre tres hipótesis con qué las distingue.
Un asistente ejecutó algo a partir de contenido de un registro	Los datos de operación se trataron como fiables y el asistente tenía permisos de escritura	Trata registros, errores y tiquetes como entrada no fiable; solo lectura por defecto y ninguna acción irreversible sin persona.
Aprobar la sugerencia se ha vuelto automático	Quien aprueba solo ve la recomendación, no la evidencia ni el coste del error	Presenta evidencia, qué se comprobó y qué no, y qué ocurre si la propuesta es errónea; un botón sin contexto es teatro de control.
El equipo responde peor a los casos raros que antes	Complacencia: el sistema resuelve lo habitual y nadie practica lo difícil	Haz ensayos periódicos con el asistente desactivado y manten procedimientos legibles por humanos.

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 señales gana la detección de anomalías y en cuáles pierde frente a un umbral?
2. ¿Por qué una anomalía es buena pista y mala alerta?
3. ¿Qué necesita la correlación para agrupar bien y qué riesgo introduce?
4. ¿Qué distingue proponer, preparar y ejecutar, y dónde está el límite?
5. ¿Qué obligación evitó que el equipo actuara sobre hipótesis sin comprobar?

Referencias

- Google (2018). The Site Reliability Workbook, cap. «Alerting on SLOs» — por qué se alerta sobre síntomas. <https://sre.google/workbook/alerting-on-slos/>
- Parasuraman, R. y Riley, V. (1997). Humans and automation: use, misuse, disuse, abuse. <https://journals.sagepub.com/doi/10.1518/001872097778543886>
- AWS (2024). CloudWatch anomaly detection. <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatchAnomalyDetection.html>
- Microsoft (2024). Azure Monitor: dynamic thresholds and smart detection. <https://learn.microsoft.com/azure/azure-monitor/alerts/alerts-dynamic-thresholds>
- OWASP (2025). Top 10 for LLM applications — entrada no fiable y permisos excesivos. <https://owasp.org/www-project-top-10-for-large-language-model-applications/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Evaluación — 263 AIOps, automatización asistida y límites humanos

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 aiops-control 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.

264 — Proyecto: centro de operaciones de CloudShop

Parte: 21 — Operación cloud, automatización y respuesta a incidentes Nivel: avanzado · Horas estimadas: 8
Laboratorio: capstone · Estado: EXECUTABLECORE

Propósito

Construir el centro de operaciones de CloudShop con todo lo de la parte 21 y comprobarlo. La clase da el encargo, el orden, el entregable y las pruebas negativas. Y cierra la parte 21: corrige las cinco predicciones de la clase 252 —dos acertadas y tres a medias—, actualiza el recuento de leyes, añade la ley 30 y escribe la hipótesis de la parte 22.

Resultados de aprendizaje

Al finalizar podrás:

1. Construir una operación completa en el orden que evita rehacer.
2. Comprobar con las pruebas negativas acumuladas de la parte.
3. Medir la operación con cifras que resistan una revisión.
4. Corregir las cinco predicciones de la clase 252 con evidencia.
5. Escribir la hipótesis de la parte 22 en forma refutable.

Conceptos centrales

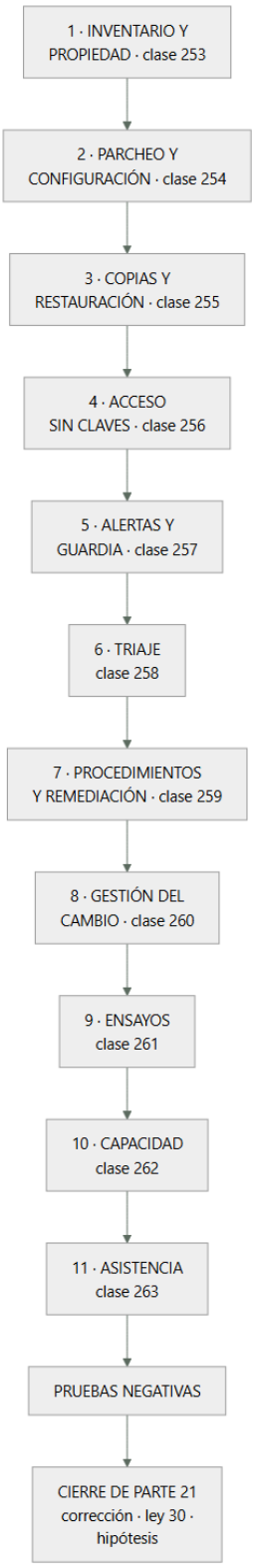
Concepto	Comprensión verificable
centro de operaciones	El conjunto de inventario, señales, guardia, procedimientos, cambio y ensayos que hace operable un sistema.
ley 30	Toda tarea manual lleva un juicio que nadie escribió; automatizarla lo elimina sin avisar.
orden por dependencia	Inventario primero, automatización al final. No se puede automatizar lo que no se sabe que existe.
prueba negativa de parte	Comprobación acumulada de las once clases, ejecutada sobre la operación entera.
trabajo repetitivo	Tarea manual, sin valor duradero, que crece con el tamaño del sistema.
hipótesis de parte	Afirmación refutable escrita antes de estudiar, que la parte siguiente corrige con evidencia.

Modelo mental

Operar es controlar cambios bajo incertidumbre: inventario, señal, diagnóstico, acción reversible y aprendizaje deben formar un ciclo cerrado.

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 orden y el entregable

El encargo. CloudShop opera 41 servicios en tres nubes, con un equipo de plataforma de nueve personas y equipos de producto que despliegan solos. Hay que dejar la operación en un estado en que una persona que entra hoy pueda estar de guardia en seis semanas.

El orden, por dependencia:

```
1  INVENTARIO Y PROPIEDAD                      clase 253
   qué hay, en qué cuenta, de quién es, para qué sirve
   → sin esto, todo lo demás actúa sobre una parte del
     sistema y nadie sabe cuál

2  PARCHEO Y CONFIGURACIÓN                     clase 254
   imágenes construidas, no retocadas; configuración
   declarada y reconciliada

3  COPIAS Y RESTAURACIÓN                       clase 255
   copias inmutables, y restauración PROBADA con reloj

4  ACCESO SIN CREDENCIALES PERMANENTES         clase 256
   sesiones auditadas, elevación temporal, sin claves
   compartidas

5  ALERTAS Y GUARDIA                           clase 257
   se alerta por síntomas del usuario; rotación
   sostenible; roles separados

6  TRIAJE                                      clase 258
   línea de cambios única y orden de descarte

7  PROCEDIMIENTOS Y REMEDIACIÓN                clase 259
   ejecutables; automáticos solo donde procede

8  GESTIÓN DEL CAMBIO                          clase 260
   lotes pequeños, vuelta atrás probada, sin comité

9  ENSAYOS                                     clase 261
   de mesa hacia arriba, con hipótesis

10 CAPACIDAD Y CUOTAS                          clase 262
   recurso limitante, codo medido, inventario de cuotas

11 ASISTENCIA AUTOMATIZADA                     clase 263
   reúne y ordena; no concluye ni ejecuta
```

Y por qué este orden y no otro:

```
los pasos 1 a 4 son el ESTADO del sistema
los pasos 5 a 8 son la RESPUESTA
los pasos 9 a 11 son la MEJORA

→ y hacer la respuesta sin el estado produce guardias que
  no pueden actuar
→ y hacer la mejora sin la respuesta produce ensayos que
  descubren lo obvio
→ intentar el paso 7 antes del 6 es el error más común: se
  automatiza sin saber diagnosticar
```

El entregable, que es lo que se evalúa:

```
1  inventario generado automáticamente, con dueño por
   recurso y cobertura medida
2  la línea de cambios única, de todas las fuentes
3  el catálogo de alertas, con la prueba de accionabilidad
   pasada
4  el calendario de guardia con la carga por turno medida
5  los procedimientos, en grado 2 o 3, con fecha de última
   ejecución
6  el inventario de automatizaciones, con condiciones,
   límite e interruptor
7  las cuatro métricas de entrega, medidas
8  el inventario de cuotas, por cuenta y región, con
   alertas
9  el registro de ensayos, con hipótesis y acciones
   cerradas
10 y el registro de decisiones: qué NO se automatizó y por
   qué
```

2. Las pruebas negativas de la parte

Cada clase dejó una comprobación que falla en la mayoría de las organizaciones. Aquí se ejecutan todas sobre el sistema entero.

```
clase 253  coge 20 recursos al azar del inventario
           → ¿los 20 tienen dueño identificable HOY?
           → y a la inversa: coge 20 recursos de la
             consola; ¿están en el inventario?

clase 254  coge una máquina de producción
           → ¿cuánto hace que se creó su imagen?
           → ¿hay algo instalado a mano en ella?

clase 255  RESTAURA una copia, con reloj
           → ¿cuánto tardó y qué se perdió?
           → ¿la copia sobrevive a quien tiene permisos de
             administrador?

clase 256  ¿queda alguna credencial permanente?
           → busca claves de acceso con más de 90 días
           → ¿alguien puede entrar sin dejar rastro?

clase 257  coge las 20 últimas alertas
           → ¿qué hizo quien las recibió?
           → si en más de 3 la respuesta es «nada»,
             el catálogo está roto

clase 258  ¿existe la línea de cambios única?
           → pide «todos los cambios de la última hora»
           → si hay que mirar en más de un sitio, no existe

clase 259  coge 5 procedimientos al azar y EJECÚTALOS
           → ¿cuántos funcionan íntegros?

clase 260  ¿cuál es la mediana de cambios por despliegue?
           → ¿cuándo se ejecutó la última vuelta atrás?
           → ¿qué porcentaje de cambios se declara urgente?

clase 261  retira una instancia de producción, ahora
           → ¿pasa lo que la hipótesis dice?

clase 262  «si el tráfico se duplica, ¿qué se rompe
           primero y en cuánto tiempo?»
           → y comprueba las cuotas de la región de
             conmutación

clase 263  coge una conclusión del asistente
           → ¿enlaza a la evidencia?
           → ¿dice qué fuentes no consultó?
```

Y la prueba que resume la parte entera:

```
LA PRUEBA DE LAS SEIS SEMANAS
  coge a alguien que entró hace seis semanas
  ponle un incidente de mesa                  clase 261
  y observa
    ¿encuentra el inventario y el dueño?
    ¿encuentra la línea de cambios?
    ¿encuentra el procedimiento y funciona?
    ¿tiene los permisos para ejecutarlo?
    ¿sabe a quién escalar y cuándo?
    ¿sabe qué comunicar y a quién?

→ si falla cualquiera, la operación depende de las
  personas que ya estaban
→ y eso es exactamente lo que se estaba intentando
  evitar
```

3. Corrección de las cinco predicciones de la clase 252

La corrección, con evidencia y sin adornos.

```
1. «los problemas de operación no son de herramientas sino
```

de INVENTARIO Y DE DUEÑO: la mayoría de los hallazgos serán cosas que existen y que nadie sabía que existían»

CORRECTA, Y MÁS AMPLIA DE LO PREVISTO. Acertamos la dirección y nos quedamos cortos en el alcance: lo desconocido no fueron solo recursos. Fueron cuotas —la región de conmutación permitía el 3,2 % de la capacidad que el plan daba por hecha—, procedimientos —19 de 34 rotos—, permisos —el rol de guardia no podía ejecutar el paso 4—, y comportamientos —una caché «temporal» de sesiones en memoria llevaba siete meses. El inventario que faltaba no era el de máquinas: era el de SUPUESTOS.

2. «la práctica que más diferencia producirá será la más aburrida: retirar. Y será la que menos se haga»

   FALLIDA EN LA PRIMERA MITAD, CORRECTA EN LA SEGUNDA. Retirar produjo mucho: 245 alertas eliminadas de 412, nueve procedimientos obsoletos, un 22 % del tráfico al quitar el sondeo del navegador. Pero la práctica que más diferencia produjo, medida, fue otra igual de aburrida: REDUCIR EL TAMAÑO DEL LOTE. Pasar de 14 cambios por despliegue a 1 llevó el tiempo de recuperación de 47 a 11 minutos y la tasa de fallo del 1,8 % al 0,6 %. Y la segunda mitad sí: nadie pidió nunca ninguna de las dos.

3. «la automatización de la remediación fracasará donde el procedimiento no estuviera antes escrito y probado»

   CORRECTA PERO INCOMPLETA, Y LO QUE FALTABA IMPORTA MÁS. Sí fracasó donde no había procedimiento probado. Pero la remediación que causó el incidente —retirar instancias no sanas— tenía el procedimiento escrito, probado y ejecutado a mano decenas de veces. Falló porque lo que una persona hace al ver que fallan nueve de once no está escrito en ninguna parte: es el juicio de PARAR. Y ese juicio se pierde al automatizar sin que nadie lo note.

4. «el tramo que dominará el tiempo total no será diagnosticar ni arreglar: será DECIDIR y COMUNICAR»

   CORRECTA Y CONFIRMADA CON CIFRAS. Un incidente resuelto técnicamente en 17 minutos consumió 1 h 21 de comunicación posterior. Y en el ensayo sin las personas clave, de los 31 minutos hasta conmutar, 14 fueron buscando el permiso y 9 esperando a quien podía autorizarlo: 23 de 31 minutos decidiendo, no diagnosticando.

5. «más de la mitad del trabajo repetitivo se podrá eliminar retirando o corrigiendo la causa, en vez de automatizándolo; y el equipo intentará automatizarlo primero»

   FALLIDA EN LA CIFRA, ACERTADA EN LA CONDUCTA. Del trabajo repetitivo identificado, el 38 % se eliminó en la causa y el 62 % se automatizó: por debajo de la mitad que predijimos. Y la conducta, exacta: en los tres casos donde acabó eliminándose la causa, la primera propuesta del equipo había sido automatizar. La fuga que se reiniciaba cada seis horas tardó cuatro meses en arreglarse, y solo porque el contador de actuaciones la delató.

Marcador: dos correctas, tres a medias. Y el fallo de la tercera es el que obliga a escribir una ley: acertamos que la automatización fracasa donde falta el procedimiento, y no vimos que también fracasa donde el procedimiento está completo, porque lo que no está escrito no es el paso sino la decisión de no darlo.

4. Recuento de leyes, ley 30 e hipótesis de la parte 22

El recuento de leyes, cerrada la parte 21.

```
ley 13  lo que no se mira deja de funcionar en silencio        66
ley 15  la señal existe y nadie la mira                        56
ley 22  un procedimiento nunca ejecutado no funciona           53
ley 14  el coste se decide al crear, no al pagar               39
ley 16  un control que estorba se rodea                        38
ley 20  lo que no tiene dueño se filtra y se desperdicia       37
ley 25  lo provisional sobrevive a su motivo                   33
ley 21  el acoplamiento vive en quién escribe                  33
ley 26  el valor por defecto sirve a la demostración           27
ley 23  la capacidad la limita lo que ya se mantiene           22
ley 17  se optimiza la medida, no el objetivo                  19
ley 24  lo que no está en el diagrama no se analiza            16
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     10
ley 29  un fallo de datos no da error: da otro número           8
ley 30  automatizar elimina un juicio que nadie escribió        6
```

Y la ley que la parte 21 obliga a escribir:

```
LEY 30
  toda tarea manual lleva un juicio que nadie escribió;
  automatizarla lo elimina sin avisar

apariciones en esta parte                                      6
  clase 258   la pregunta «¿es una de N?» no estaba en
              ningún procedimiento y resolvió en 22 minutos
              un caso de nueve días
  clase 259   retirar instancias no sanas era correcto; lo
              que faltaba era la decisión de no hacerlo
              cuando fallan nueve de once
  clase 259   en modo sombra, dos de siete automatizaciones
              habrían actuado en momentos en que una
              persona no lo habría hecho
  clase 261   sin las dos personas que más sabían, la misma
              conmutación tardó 26 minutos más: la
              diferencia era juicio, no conocimiento
  clase 262   escalar el servicio era la respuesta correcta
              y multiplicó las conexiones contra el recurso
              limitante
  clase 263   el asistente concluyó sin decir qué no había
              mirado; una persona habría dicho «no tengo
              acceso a los cambios de red»

y lo que la distingue de la ley 22
  la 22 dice que un procedimiento no ejecutado no funciona
  la 30 dice algo peor: que un procedimiento PERFECTAMENTE
  ejecutado a mano durante años sigue estando incompleto,
  porque la parte que lo hace seguro nunca se escribió
  → y por eso el modo sombra vale tanto: es la única forma
    de descubrir el juicio ausente antes de perderlo
                                                  clase 259
```

La hipótesis de la parte 22 (clases 265 a 276, especialización, certificación y carrera), escrita antes de estudiarla:

```
1. las certificaciones correlacionarán con conseguir
   entrevistas y no con resolver problemas; y el hueco
   estará justo donde no se puede examinar por opción
   múltiple: decidir con información incompleta
                                                    ley 17

2. la especialización que peor envejecerá será la atada a
   un producto, y la que mejor, la atada a una RESTRICCIÓN
   —red, datos, coste, seguridad—, porque las restricciones
   sobreviven a los productos que las gestionan

3. buena parte del consejo de carrera tratará de
   visibilidad y no de destreza; será incómodo y será
   cierto, y lo medible será que lo que predice una
```

```
   promoción no es lo que predice resolver un incidente

4. lo que el mercado paga en multinube no será saber tres
   nubes: será saber una a fondo y poder TRADUCIR; y quien
   sepa las tres por encima rendirá peor que quien sepa una
   y entienda el modelo                          ley 24

5. y una refutable con cifra: **más de la mitad de lo
   enseñado en este programa 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
```

Y el cierre de la parte 21: de once clases, lo más caro no fue ningún fallo técnico: fue que 19 de 34 procedimientos estaban rotos y nadie podía saberlo, porque un documento que nunca se ejecuta no tiene forma de avisar de que ha envejecido. La parte 22 cambia de plano: deja el sistema y mira a quien lo opera. Empieza por el mapa de especializaciones y qué exige cada una. Es la clase 265.

Ejemplo trabajado

El centro de operaciones de CloudShop, resuelto. Lo que sigue son las decisiones con su motivo, el resultado de las once pruebas negativas, y las cifras de la operación a los quince meses.

Las decisiones, por capa.

```
INVENTARIO Y PROPIEDAD                          clase 253
  generado desde las tres nubes cada 6 horas
  dueño obligatorio por etiqueta; sin dueño, se marca y a
    los 30 días se apaga en entornos no productivos
  cobertura de dueño       61 % → 99,4 %
  recursos sin uso hallados y retirados      1.107

PARCHEO Y CONFIGURACIÓN                         clase 254
  imágenes construidas en cadena, con antigüedad máxima
    de 30 días
  configuración declarada y reconciliada; la deriva se
    corrige o se alerta
  máquinas con cambios manuales    23 → 0

COPIAS Y RESTAURACIÓN                           clase 255
  copias inmutables, en cuenta separada          clase 219
  restauración probada mensualmente, con reloj
  tiempo real de restauración de la base de pedidos
    creído 40 min · medido 3 h 10 · tras trabajo 52 min

ACCESO                                          clase 256
  cero credenciales permanentes; sesiones registradas
  elevación temporal con motivo
  claves de acceso con más de 90 días     41 → 0

ALERTAS Y GUARDIA                               clase 257
  alertas    412 → 167 · accionables 8 % → 85 %
  guardia de 6 personas por rotación
  roles separados: coordina, arregla, comunica, anota

TRIAJE                                          clase 258
  línea de cambios única, de 7 fuentes
  consultas de triaje guardadas y enlazadas

PROCEDIMIENTOS                                  clase 259
  34 → 25, todos en grado 2; 8 en grado 3
  con condiciones, límite, interruptor y contador

CAMBIO                                          clase 260
  comité disuelto; comprobaciones automáticas
  94 % de cambios estándar; lote mediano de 1

ENSAYOS                                         clase 261
  28 en 15 meses; 134 hallazgos, 68 % organizativos

CAPACIDAD                                       clase 262
  recurso limitante identificado por servicio
```

```
  inventario de cuotas automático, 63 relevantes
  vertido de carga por prioridad, probado

ASISTENCIA                                   clase 263
  reúne, resume y propone; no concluye ni ejecuta
```

Las once pruebas negativas, ejecutadas.

```
253  20 recursos con dueño identificable       11/20      20/20
     20 recursos de consola en inventario      14/20      20/20
254  antigüedad de la imagen en producción  312 días    19 días
     algo instalado a mano                       sí         no
255  restauración con reloj                  3 h 10     52 min
     copia sobrevive a administrador             no         sí
256  claves permanentes > 90 días                41          0
     acceso sin rastro                           sí         no
257  de 20 alertas, cuántas hicieron actuar    2/20      17/20
258  línea de cambios en un solo sitio           no         sí
259  de 5 procedimientos, cuántos funcionan     1/5        5/5
260  mediana de cambios por despliegue            14          1
     última vuelta atrás ejecutada         14 meses     6 días
     cambios declarados urgentes               12 %      1,9 %
261  retirar una instancia: ¿pasa lo previsto?   no         sí
262  «si el tráfico se duplica, ¿qué se rompe?»
                                        sin respuesta  medido
     cuotas de la región de conmutación        3,2 %      100 %
263  conclusión del asistente con evidencia      no         sí
```

Y la prueba de las seis semanas, ejecutada dos veces:

```
mes 4, con una persona incorporada 6 semanas antes
  encuentra inventario y dueño                       sí
  encuentra la línea de cambios                      sí
  encuentra el procedimiento y funciona              sí
  tiene los permisos                                 NO
  sabe a quién escalar                               sí
  sabe qué comunicar                                 NO
  → dos fallos; ambos corregidos

mes 11, con otra persona
  las seis, sí
  tiempo hasta mitigar en el ensayo             11 min
  (la media del equipo veterano era de 9)
```

Las cifras de la operación, a los quince meses.

```
DISPONIBILIDAD Y RESPUESTA
incidentes de gravedad alta            34/año      13/año
tiempo medio hasta mitigar             1 h 20      14 min
tiempo medio de recuperación           47 min      11 min
incidentes sin causa identificada       26/34        2/13

CAMBIO
frecuencia de despliegue           3,1/semana   47/semana
tiempo de entrega                     6,4 días      3,2 h
tasa de fallo por cambio                1,8 %      0,6 %

PERSONAS
interrupciones de guardia por turno        9,4        2,6
fuera de horario, por turno                3,1        0,7
rotación mínima (personas)                   4          6
abandonos del equipo de plataforma
  en el periodo                              3          0

TRABAJO
situaciones resueltas sin persona        0/mes      73/mes
trabajo repetitivo (h/semana/persona)     14,2        4,8
  eliminado en la causa                        38 %
  automatizado                                 62 %

COSTE
coste de gobierno del cambio        546 h/año     72 h/año
recursos sin uso retirados                  1.107
ahorro anual asociado                  41.200 USD
```

Y el dato que el equipo puso primero en la revisión:

```
abandonos del equipo de plataforma
  en los 15 meses anteriores                          3
  en los 15 meses del proyecto                        0

y lo que las entrevistas de salida habían dicho
  «la guardia es insostenible»                        3 de 3

→ de 3,1 a 0,7 interrupciones nocturnas por turno
→ y esa es la métrica que la dirección entendió sin
  explicación
```

Y el registro de decisiones: qué NO se automatizó y por qué.

```
conmutación de región        acción no segura si el
                             diagnóstico falla
restauración desde copia     no reversible
desviar tráfico entre        impacto amplio; decisión de
proveedores                  negocio
desactivar el pago           decisión de negocio
ampliar cuota de la base     coste; requiere aprobación
comunicación con clientes    nunca                clase 257
y cualquier acción propuesta
por el asistente             nunca sin persona    clase 263

→ y este registro es el entregable que más se consultó
  después
→ porque responde a la pregunta que vuelve cada trimestre:
  «¿por qué esto todavía lo hace una persona?»
```

Lo que salió mal durante el proyecto, que también es entregable:

```
1  la primera remediación automática convirtió una
   degradación en una caída total en 3 minutos
                                                clase 259
2  el ensayo 4 se abortó por errores de usuario que la
   hipótesis no preveía                         clase 261
3  el asistente costó 26 minutos por concluir sin declarar
   qué no había mirado                          clase 263
4  la migración de esquema se detuvo en la semana 2 por 41
   divergencias por 100.000                     clase 260
5  y el detector de anomalías, adoptado tal cual, habría
   producido 4.180 avisos al día                clase 263

→ los cinco se descubrieron con radio acotado y con gente
  delante
→ ninguno llegó a un incidente nocturno
→ y ese es el argumento entero de la parte 21
```

La lección que esta clase deja: la operación mejoró en las doce métricas, pero la que convenció a la dirección no fue ninguna de disponibilidad: fue cero abandonos frente a tres, con las interrupciones nocturnas por turno de 3,1 a 0,7. Y de los cinco fallos del proyecto, los cinco ocurrieron con radio acotado, en horario laboral y con observadores, que es exactamente la diferencia entre un ensayo y un incidente.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-21-cloud-operations-automation/264-proyecto-centro-de-operaciones-de-cloudshop/
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 cloudshop-operations 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 cloudshop-operations 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 automatiza la respuesta antes de saber diagnosticar	Se saltó el orden: la respuesta se montó sin el estado del sistema debajo	Inventario, parcheo, copias y acceso primero; luego alertas y triaje; la automatización va después, no antes.
La operación funciona solo si están las personas de siempre	El conocimiento vive en las cabezas y no en inventario, procedimientos y permisos	Aplica la prueba de las seis semanas: si alguien recién incorporado no puede responder un incidente de mesa, sabes exactamente qué falta.
Todas las métricas mejoran y el equipo sigue quemado	Se midió el sistema y no las interrupciones por turno	Mide interrupciones de guardia por turno y fuera de horario; es la métrica que predice abandonos y la que entiende la dirección.
Nadie recuerda por qué una tarea sigue siendo manual	No se registró la decisión de no automatizarla	Manten un registro de qué no se automatizó y por qué; es lo que responde a la pregunta que vuelve cada trimestre.
Una automatización correcta hizo daño en una situación que nadie previó	El juicio de cuándo no actuar nunca se escribió porque la persona lo aplicaba sin pensarlo	Estrena en modo sombra y revisa cada caso en que habría actuado; es la única forma de encontrar el juicio ausente antes de perderlo.
El proyecto de operación se estanca sin resultados visibles	Se empezó por lo llamativo en vez de por el inventario y la propiedad	El inventario con dueño es el primer entregable y el que desbloquea todo lo demás; sin él, cada mejora actúa sobre una parte desconocida del sistema.

■ 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 el inventario va antes que la automatización y qué pasa si se invierte?
2. ¿Qué comprueba la prueba de las seis semanas y qué revela cuando falla?

3. ¿Cuál de las cinco predicciones falló y qué ley obligó a escribir?
4. ¿Qué distingue la ley 30 de la ley 22?
5. ¿Qué debe contener el registro de lo que no se automatizó?

Referencias

- Beyer, B. y otros (2018). The Site Reliability Workbook — operación completa de extremo a extremo. <https://sre.google/workbook/table-of-contents/>
- Forsgren, N., Humble, J. y Kim, G. (2018). Accelerate — las cuatro métricas de entrega. <https://itrevolution.com/product/accelerate/>
- AWS (2024). Operational Excellence Pillar. <https://docs.aws.amazon.com/wellarchitected/latest/operational-excellence-pillar/welcome.html>
- Microsoft (2024). Azure Well-Architected Framework: Operational Excellence. <https://learn.microsoft.com/azure/well-architected/operational-excellence/>
- Google Cloud (2024). Architecture Framework: Operational excellence. <https://cloud.google.com/architecture/framework/operational-excellence>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 21 en PDF · Manual integral

Anterior	Índice	Siguiente
← 263 · AIOps, automatización asistida y límites humanos	Parte 21 · Programa	265 · Ruta Cloud Engineer y mapa de competencias →

Evaluación — 264 Proyecto: centro de operaciones de CloudShop

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 cloudshop-operations 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.