Multi-Cloud Engineering Program

Parte 11 — Seguridad, gobierno, cumplimiento y FinOps

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	11 de 23
Clases	133–144
Evaluaciones	12
Horas estimadas	52

Índice

Parte 11 — Seguridad, gobierno, cumplimiento y FinOps

← Parte 10 · Índice completo · Parte 12 →

Descargar: Esta parte en PDF · Manual integral

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

Construir guardrails que hagan segura y económicamente sostenible la autonomía.

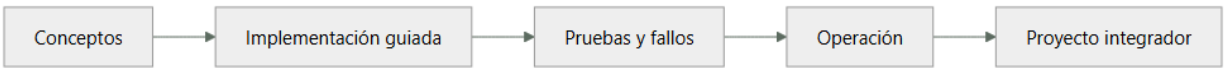
Resultados de la parte

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

Gobernar no significa aprobar cada cambio: significa codificar límites, evidencia y responsabilidades para que los equipos puedan avanzar con seguridad.

Secuencia



Clases

ID	Clase	Laboratorio	Horas
133	Zero Trust y defensa en profundidad	security	4
134	Mínimo privilegio, acceso temporal y separación de funciones	iam	4
135	Segmentación, perímetro, WAF, DDoS y egress	network	4
136	Cifrado, KMS, HSM, rotación y envelope encryption	security	4
137	Gestión de secretos y credenciales de workloads	security	4
138	Vulnerabilidades, imágenes y cadena de suministro	supply-chain	4
139	CSPM, postura, policy as code y remediación	governance	4
140	Threat modeling con STRIDE y attack paths	security	4
141	Cumplimiento, residencia, privacidad y evidencia	compliance	4
142	FinOps: showback, chargeback, budgets y anomalías	finops	4
143	Optimización de costo, capacidad y sostenibilidad	finops	4
144	Proyecto: landing zone con guardrails	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

- Practical Cloud Security — Chris Dotson.
- Cloud Security Handbook — Eyal Estrin.
- Cloud FinOps — Storment y Fuller.

133 — Zero Trust y defensa en profundidad

Parte: 11 — Seguridad, gobierno, cumplimiento y FinOps Nivel: avanzado · Horas estimadas: 4 Laboratorio: security · Estado: EXECUTABLECORE

Propósito

Abrir la parte del control con el cambio de modelo que la hace posible: dejar de conceder confianza por el lugar del que viene una petición. La clase separa la idea de los productos que la venden, explica por qué el modelo de perímetro convierte el primer equipo comprometido en la red entera, y desarrolla el criterio que distingue una defensa en profundidad real de cinco controles decorativos: que las capas sean independientes. Y termina con la pregunta que ordena todo el trabajo: si roban ahora mismo esta credencial, ¿hasta dónde se llega?

Resultados de aprendizaje

Al finalizar podrás:

1. Enunciar qué se verifica en cada petición y por qué no basta el origen.
2. Explicar por qué el modelo de perímetro falla y qué lo sustituye.
3. Distinguir capas de defensa independientes de capas que comparten un único punto.
4. Calcular el alcance de un compromiso desde cada punto de entrada.
5. Ordenar la adopción por lo que de verdad detiene ataques reales.

Conceptos centrales

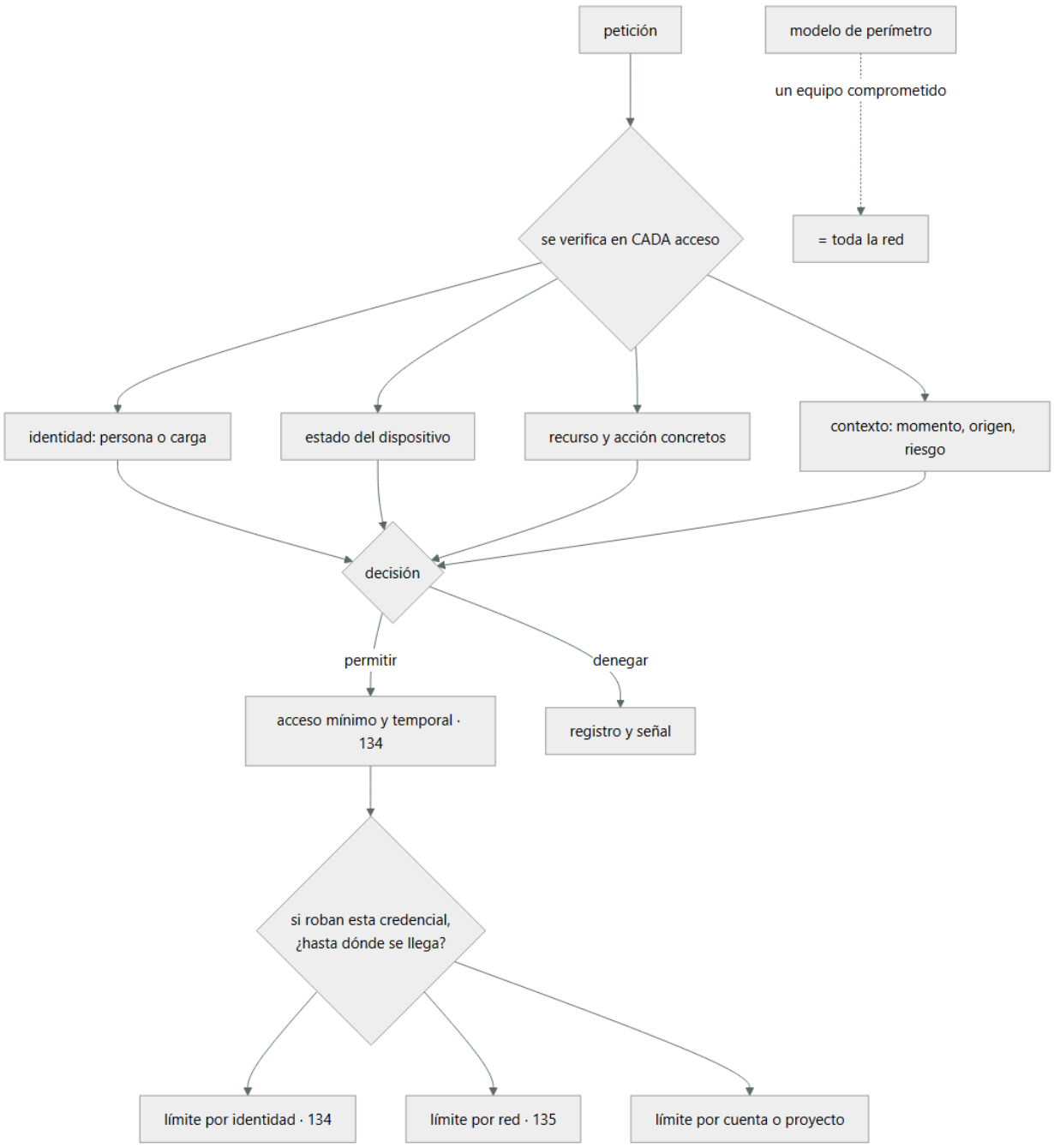
Concepto	Comprensión verificable
confianza cero	Ninguna petición se acepta por venir de un sitio determinado. Se verifica identidad, permiso y contexto en cada acceso.
modelo de perímetro	Dividir el mundo en dentro y fuera, y confiar en lo de dentro. Su consecuencia es que un solo equipo comprometido da acceso a todo.
movimiento lateral	Avance del atacante de un sistema a otro una vez dentro. Es donde ocurre la mayor parte del daño.
independencia de capas	Que el fallo de una no implique el de las demás. Cinco controles que dependen del mismo sistema de identidad son un solo control.
alcance del compromiso	Todo lo que un atacante puede leer, escribir o ejecutar partiendo de una credencial concreta.
suponer la brecha	Diseñar dando por hecho que alguien ya está dentro, y limitar lo que puede hacer desde allí.

Modelo mental

Gobernar no significa aprobar cada cambio: significa codificar límites, evidencia y responsabilidades para que los equipos puedan avanzar con seguridad.

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. Por qué el perímetro dejó de funcionar

El modelo tradicional divide el mundo en dos y confía en uno de los lados:

```
fuera   no confiable: cortafuegos, filtrado, inspección
dentro  confiable: se accede a casi todo sin más comprobaciones
```

Y su consecuencia se enuncia en una línea:

```
el primer equipo comprometido dentro equivale a la red entera
```

Porque a partir de ahí el atacante ya está «dentro», y dentro no hay controles. Eso es el movimiento lateral, y es donde ocurre la mayor parte del daño de casi cualquier incidente serio.

Y hay cuatro razones por las que el modelo ya no se sostiene, aunque la anterior bastaría:

```
la carga vive en varios proveedores y en servicios gestionados
  → no hay un dentro que contenga todo
la gente trabaja desde cualquier sitio
```

```
  → el túnel privado convierte a cualquier portátil en «dentro»
las direcciones son dinámicas y efímeras
  → una regla por dirección no significa nada en contenedores
las integraciones con terceros atraviesan el perímetro por diseño
```

Lo que lo sustituye es una regla sencilla de enunciar y laboriosa de implantar:

```
ninguna petición se acepta por venir de un sitio determinado
se verifica en CADA acceso:
  quién      identidad de la persona o de la carga, demostrada
  con qué    estado del dispositivo, cuando aplique
  a qué      recurso concreto
  para qué   acción concreta
  en qué contexto   momento, origen, señales de riesgo
```

Y dos precisiones para no vender humo:

```
esto NO es un producto que se compra
  es una dirección arquitectónica que se recorre por partes

la red sigue importando
  no se deja de segmentar: se deja de CONFIAR por estar segmentado
  → la red pasa de ser el control a ser una capa más     clase 135
```

Y la pieza que hace posible todo lo demás es la que este programa lleva construyendo desde la clase 054: identidad de carga de trabajo. Un servicio se autentica por lo que es —una identidad emitida y verificable—, no por la dirección desde la que llama.

2. Capas que de verdad son capas

La defensa en profundidad consiste en poner varios controles de modo que hagan falta varios fallos para que haya daño. Y su fallo típico no es tener pocas capas, sino tenerlas mal:

```
control 1   autenticación en la puerta de entrada
control 2   permisos del servicio
control 3   política de red que permite el tráfico
control 4   cifrado con claves gestionadas
control 5   registro de auditoría

y los cinco se apoyan en el mismo sistema de identidad
→ quien lo comprometa pasa las cinco a la vez
→ no son cinco capas: es una
```

De ahí el criterio:

```
dos capas son independientes si el fallo de una no implica el de la otra
→ y se comprueba preguntando: ¿qué comprometería a las dos?
```

Y las dimensiones sobre las que sí se consiguen capas independientes:

```
IDENTIDAD      quién puede pedirlo                    clase 134
RED            desde dónde se puede alcanzar          clase 135
DATO           cifrado, con claves de otro dominio    clase 136
APLICACIÓN     validación y autorización de negocio
DETECCIÓN      registro, alertas y respuesta          parte 10
ORGANIZACIÓN   separación de funciones y aprobaciones  clase 134
```

Y una comprobación práctica sobre cualquier sistema:

```
si un atacante consigue la credencial de la aplicación,
¿qué le impide leer la base de datos entera?
  «los permisos»            → una capa
  «y además la red»         → dos, si la red no la controla la misma identidad
  «y el cifrado por campo»  → tres, si la clave no la puede pedir esa identidad
```

La última condición es la que suele fallar: la aplicación puede descifrar, así que el cifrado no es una capa frente a ella. Sí lo es frente a quien acceda al almacenamiento por otro camino, que es un escenario distinto y también real.

Y el peor error de este apartado, que aparece constantemente:

```
confundir NÚMERO de controles con profundidad
→ veinte reglas que dependen de lo mismo protegen como una
```

Y su contrario, igual de dañino: una capa que estorba y se rodea no cuenta. Es la ley 16, que esta parte va a ver muchas veces.

3. Suponer la brecha, y medirla

El ejercicio más útil de esta clase no es un diagrama: es una pregunta con respuesta medible.

```
si roban AHORA MISMO esta credencial, ¿hasta dónde se llega?
```

Y se responde punto por punto, enumerando los orígenes reales:

```
el portátil de alguien de desarrollo
la identidad de la canalización                    clase 098
la identidad de un contenedor en producción        clase 054
una clave de API de un socio                       clase 118
la cuenta de un proveedor externo con acceso
un servicio expuesto a internet con una vulnerabilidad
```

Y para cada uno se enumera lo alcanzable, sin adornos:

```
¿qué datos puede leer?
¿qué puede modificar o borrar?
¿puede conseguir OTRA credencial desde ahí?     ← lo más importante
¿puede llegar a producción desde un entorno inferior?
¿queda registro de lo que haga?
¿cuánto tardaría alguien en enterarse?
```

La tercera pregunta es la que decide el alcance real: la mayoría de los compromisos graves no vienen de lo que la primera credencial permite, sino de la segunda que permite obtener.

Y las tres formas de acortar el alcance, en orden de eficacia:

```
1. que la credencial sea de corta vida y de un solo uso
   → nada de claves de larga duración                clase 098
2. que sus permisos sean los mínimos y no permanentes
   → nada de administradores permanentes              clase 134
3. que no pueda alcanzar por red lo que no necesita
   → segmentación con denegación por defecto          clase 135
```

Y hay una cuarta que es de las más eficaces y menos usadas: separar por cuentas o proyectos. Los límites administrativos del proveedor son la frontera más difícil de cruzar para un atacante, porque no dependen de configuración de red ni de reglas finas.

```
un entorno por cuenta o proyecto
los entornos inferiores SIN camino hacia producción
y las identidades de uno sin permisos en el otro
```

Y el orden de adopción, que importa porque casi todo el mundo empieza por el sitio equivocado:

```
1. identidad de las personas: factor resistente a suplantación
2. identidad de las cargas: sin claves de larga duración
3. permisos mínimos y sin administración permanente
4. segmentación que limite el movimiento lateral
5. detección y respuesta
```

Y la razón del orden es que los dos primeros cortan la vía de entrada más común, mientras que empezar por la red es caro y deja abierta la puerta principal.

4. El control que se rodea

Este apartado es la ley 16 aplicada a la seguridad, y conviene abrirlo aquí porque va a reaparecer en toda la parte.

```
un control que estorba acaba desactivado, rodeado o convertido en trámite
```

Y sus formas típicas, con lo que las produce:

```
el túnel privado que todo el mundo evita
  porque va lento y hay que reconectarse cada hora
  → la gente usa alternativas sin control

la excepción temporal que lleva dos años
  porque nadie puso fecha                          clases 046, 091

la cuenta administrativa compartida
  porque pedir permisos tarda tres días
  → y entonces nadie sabe quién hizo qué

el repositorio de claves en una hoja de cálculo
  porque el almacén oficial exige un procedimiento largo
```

```
la aprobación que se concede sin mirar
  porque llegan cuarenta al día                          ley 15
```

Y la conclusión operativa, que va contra la intuición: el control más seguro que nadie usa es peor que uno razonable que todo el mundo usa.

Las cuatro formas de que un control sobreviva:

```
QUE SEA EL CAMINO MÁS RÁPIDO
  pedir un acceso temporal debe tardar menos que buscar un atajo
  → es el camino asfaltado de la clase 106 aplicado a la seguridad
QUE TENGA SALIDA DECLARADA
  con motivo, responsable y caducidad que rompa algo    clase 101
QUE SE MIDA SU RODEO
  cuántas excepciones, cuántos accesos por vía alternativa
  → si sube, el control está mal diseñado, no la gente
QUE NO DEPENDA DE LA MEMORIA DE NADIE
  automático por defecto, no una lista de buenas prácticas
```

Y lo que hay que vigilar desde el primer día de esta parte:

```
excepciones vivas y su antigüedad
credenciales de larga duración que quedan
identidades con permisos amplios permanentes
accesos a producción fuera del camino habitual
cuentas compartidas
```

Y la lista de comprobación de la clase:

- ninguna autorización depende únicamente del origen de la petición
- las cargas se autentican por identidad emitida, no por dirección
- está escrito qué se verifica en cada acceso
- las capas de defensa son independientes, y se ha comprobado
- está calculado el alcance desde cada punto de entrada
- está identificado desde dónde se puede obtener OTRA credencial
- los entornos están separados por cuenta o proyecto
- los entornos inferiores no tienen camino a producción
- el camino seguro es el más rápido
- se mide cuánto se rodean los controles

Y el cierre que enlaza con la clase siguiente: de las tres formas de acortar el alcance, la que más reduce y más fricción genera es la de los permisos. Cómo se conceden los mínimos, cómo se dan solo mientras hacen falta y cómo se impide que una sola persona pueda hacerlo todo es la materia de la clase 134.

Ejemplo trabajado

CloudShop hace el ejercicio del apartado tercero antes de comprar ni configurar nada: enumerar los puntos de entrada y medir hasta dónde se llega desde cada uno. El resultado reordena por completo lo que se pensaba hacer.

El inventario de puntos de entrada.

```
1. portátil de una persona de desarrollo
2. identidad de la canalización                    clase 098
3. contenedor de producción del servicio de pedidos
4. clave de API de un socio                        clase 118
5. cuenta de un proveedor de análisis con acceso de lectura
6. servicio expuesto a internet
```

Lo que se alcanzaba desde cada uno, medido con permisos reales.

```
1. PORTÁTIL DE DESARROLLO
   túnel privado activo → red interna completa
   credenciales en el disco: 3 claves de larga duración
   acceso a producción: SÍ, mediante una de esas claves
   otra credencial obtenible: sí, la del almacén de secretos
   → ALCANCE: total

2. CANALIZACIÓN
   ya corregido en la clase 103: sin credenciales de clúster
   puede confirmar en el repositorio de entorno
   → ALCANCE: puede desplegar lo que quiera, con revisión en producción

3. CONTENEDOR DE PEDIDOS
   identidad federada, sin claves      clase 054
```

```
   base de datos de pedidos: lectura y escritura
   otras bases: NO
   red: puede alcanzar 14 de 15 servicios
   otra credencial obtenible: sí, la del proveedor de pago
   → ALCANCE: pedidos + pagos

4. CLAVE DE SOCIO
   API pública, con límites
   → ALCANCE: sus propios datos

5. PROVEEDOR DE ANÁLISIS
   lectura sobre el lago completo, incluido el subconjunto
   con datos personales     clase 104
   → ALCANCE: todos los datos históricos

6. SERVICIO EXPUESTO
   identidad propia, permisos limitados
   red: puede alcanzar 14 de 15 servicios
   → ALCANCE: punto de partida para movimiento lateral
```

Las tres conclusiones que reordenaron el plan.

El plan original era «segmentar la red», porque era lo que estaba en el presupuesto. Los números decían otra cosa:

```
1. el punto de entrada más peligroso era un PORTÁTIL, no un servidor
   tres claves de larga duración en disco daban acceso total

2. el multiplicador era la SEGUNDA credencial
   4 de 6 puntos permitían obtener otra credencial
   → sin eso, ningún punto llegaba lejos

3. la red importaba, y menos de lo esperado
   14 de 15 servicios alcanzables desde cualquier sitio era malo
   pero por sí sola no daba acceso a datos: hacía falta una credencial
```

Lo que se hizo, en el orden que los datos dictaron.

```
MES 1-2  identidad de las personas
  factor resistente a suplantación, obligatorio
  fin de las claves de larga duración en portátiles
  acceso a producción solo mediante concesión temporal (clase 134)

MES 2-3  identidad de las cargas
  las 3 credenciales estáticas restantes migradas a federación
  el proveedor de pago pasó a clave rotada automáticamente

MES 3-5  separación por cuentas
  un proyecto por entorno; sin camino de dev a producción
  el proveedor de análisis, movido a una cuenta de solo lectura
  con datos anonimizados

MES 5-8  segmentación
  denegación por defecto entre servicios (clase 135)
```

El mismo ejercicio, repetido a los ocho meses.

```
portátil de desarrollo                    total          nada sin concesión
                                                         temporal aprobada
canalización                          desplegar todo     igual, con revisión
contenedor de pedidos                pedidos + pagos     solo pedidos
clave de socio                          sus datos           sus datos
proveedor de análisis              todo el histórico    subconjunto anonimizado
servicio expuesto                14 de 15 servicios       2 de 15 servicios

puntos desde los que se obtiene otra credencial   4 de 6        0 de 6
```

La última fila es la que más cambió el riesgo: ninguna credencial permite ya conseguir otra.

La comprobación de independencia de capas.

Se revisaron los controles del servicio de pedidos:

```
control                              ¿de qué depende?
autenticación en la puerta            sistema de identidad
permisos del servicio                 sistema de identidad
política de red                       controlador de red
```

```
cifrado en reposo                    servicio de claves
registro de auditoría                plataforma

¿qué comprometería a la vez varias?
  el sistema de identidad → las dos primeras
  → son UNA capa, no dos
```

Y la corrección no fue añadir controles, sino separarlos:

```
la política de red pasó a no depender de etiquetas gestionadas
por la misma identidad
el acceso a las claves de cifrado se restringió a identidades
distintas de las de la aplicación, con aprobación para operaciones
masivas de descifrado
```

La ley 16, medida durante la adopción.

```
excepciones vivas                          0            31
de ellas, con fecha de caducidad           —            31
de ellas, caducadas y renovadas            —             6
cuentas compartidas                        4             0
accesos a producción por vía alternativa   no medido     2 / mes
tiempo para obtener acceso temporal        3 días       90 s
```

Y la penúltima fila es la que se vigila: dos accesos al mes por vía alternativa se investigaron uno a uno; los dos eran procedimientos de emergencia sin camino oficial, y se les creó uno.

Y la última explica por qué el resto funcionó: pedir acceso pasó de tres días a noventa segundos, y con eso nadie tuvo motivo para buscar atajos.

A los ocho meses.

```
claves de larga duración                      14             0
puntos de entrada con alcance total            1             0
puntos desde los que se obtiene otra credencial  4           0
servicios alcanzables desde uno comprometido   14 de 15      2 de 15
entornos separados por cuenta                  no             sí
camino de dev a producción                     sí             no
cuentas compartidas                            4             0
tiempo para acceso temporal a producción       3 días        90 s
capas realmente independientes                 3             5
```

La lección que esta clase abre para la parte 11: el plan original era segmentar la red, y el ejercicio de medir el alcance demostró que el punto de entrada más peligroso era un portátil con tres claves guardadas en disco. Lo que multiplicaba el daño no era la conectividad: era que cuatro de seis credenciales permitían obtener otra. Y el cambio que hizo aceptable todo lo demás no fue un control: fue reducir de tres días a noventa segundos el tiempo de pedir acceso, porque un control que estorba se rodea, y eso es exactamente lo que la hipótesis de la clase 132 predijo para esta parte.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-11-security-governance-finops/133-zero-trust-y-defensa-en-profundidad/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 modelo-zero-trust 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 modelo-zero-trust 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 solo equipo comprometido da acceso a casi todo	Modelo de perímetro: dentro no hay controles y el movimiento lateral es libre	Verifica identidad, recurso, acción y contexto en cada acceso; separa entornos por cuenta y deniega por defecto entre servicios.
Hay muchos controles y un solo fallo los atraviesa todos	Las capas dependen del mismo sistema; no son capas	Comprueba para cada par qué comprometería a los dos y construye sobre dimensiones distintas: identidad, red, dato, aplicación, detección y organización.
Se invierte en segmentar la red y los incidentes siguen entrando por credenciales	Se empezó por la capa cara antes que por la identidad	Ordena la adopción: identidad de personas, identidad de cargas, permisos mínimos, segmentación y detección.
Un compromiso pequeño acaba siendo total	Desde la primera credencial se puede obtener otra	Enumera para cada punto de entrada si permite conseguir otra credencial, y corta esa cadena antes que nada.
La gente usa vías alternativas para trabajar	Ley 16: el control oficial es más lento que el atajo	Haz que el camino seguro sea el más rápido, mide los rodeos y trata su aumento como defecto de diseño.
Se declara implantada la confianza cero porque se compró un producto	Es una dirección arquitectónica, no un producto	Mide alcance por punto de entrada antes y después; esa cifra es la que dice si algo ha cambiado.

■ 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 modelo de perímetro convierte un equipo comprometido en la red entera?
2. ¿Qué se verifica en cada acceso cuando no se confía en el origen?
3. ¿Cómo se comprueba que dos capas de defensa son independientes?

4. ¿Por qué la pregunta clave es si desde una credencial se puede obtener otra?
5. ¿En qué orden conviene adoptar los controles y por qué no se empieza por la red?

Referencias

- NIST (2020). SP 800-207: Zero Trust Architecture — principios, componentes y modelos de despliegue. <https://csrc.nist.gov/pubs/sp/800/207/final>
- Google (2025). BeyondCorp: a new approach to enterprise security — acceso sin confianza en la red. <https://cloud.google.com/beyondcorp>
- CISA (2025). Zero Trust Maturity Model — orden de adopción por pilares. <https://www.cisa.gov/zero-trust-maturity-model>
- MITRE (2025). ATT&CK: lateral movement — técnicas reales de avance dentro de una red. <https://attack.mitre.org/tactics/TA0008/>
- AWS (2025). Security pillar: apply security at all layers — independencia de capas y separación por cuentas. <https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/welcome.html>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 11 en PDF · Manual integral

Anterior	Índice	Siguiente
← 132 · Proyecto: operación SRE de CloudShop	Parte 11 · Programa	134 · Mínimo privilegio, acceso temporal y separación de funciones →

Evaluación — 133 Zero Trust y defensa en profundidad

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 modelo-zero-trust 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.

134 — Mínimo privilegio, acceso temporal y separación de funciones

Parte: 11 — Seguridad, gobierno, cumplimiento y FinOps Nivel: avanzado · Horas estimadas: 4 Laboratorio: iam · Estado: EXECUTABLECORE

Propósito

Reducir lo que se puede hacer con una credencial robada, que es la palanca que la clase 133 midió como la más eficaz. La clase sostiene tres cosas concretas: que el problema no es tener demasiados permisos, sino tenerlos siempre, y que la corrección es que caduquen por defecto; que los permisos se deducen del uso observado y no de lo que alguien imagina que hará falta; y que la revisión periódica de accesos, tal como se practica, es teatro —lo que funciona es retirar automáticamente lo que no se usa—.

Resultados de aprendizaje

Al finalizar podrás:

1. Derivar un conjunto de permisos a partir del uso real registrado.
2. Sustituir el acceso permanente por concesiones temporales.
3. Separar funciones de forma que una sola persona no complete una acción dañina.
4. Diseñar el acceso de emergencia para que exista y deje rastro.
5. Retirar permisos sin depender de que alguien los revise.

Conceptos centrales

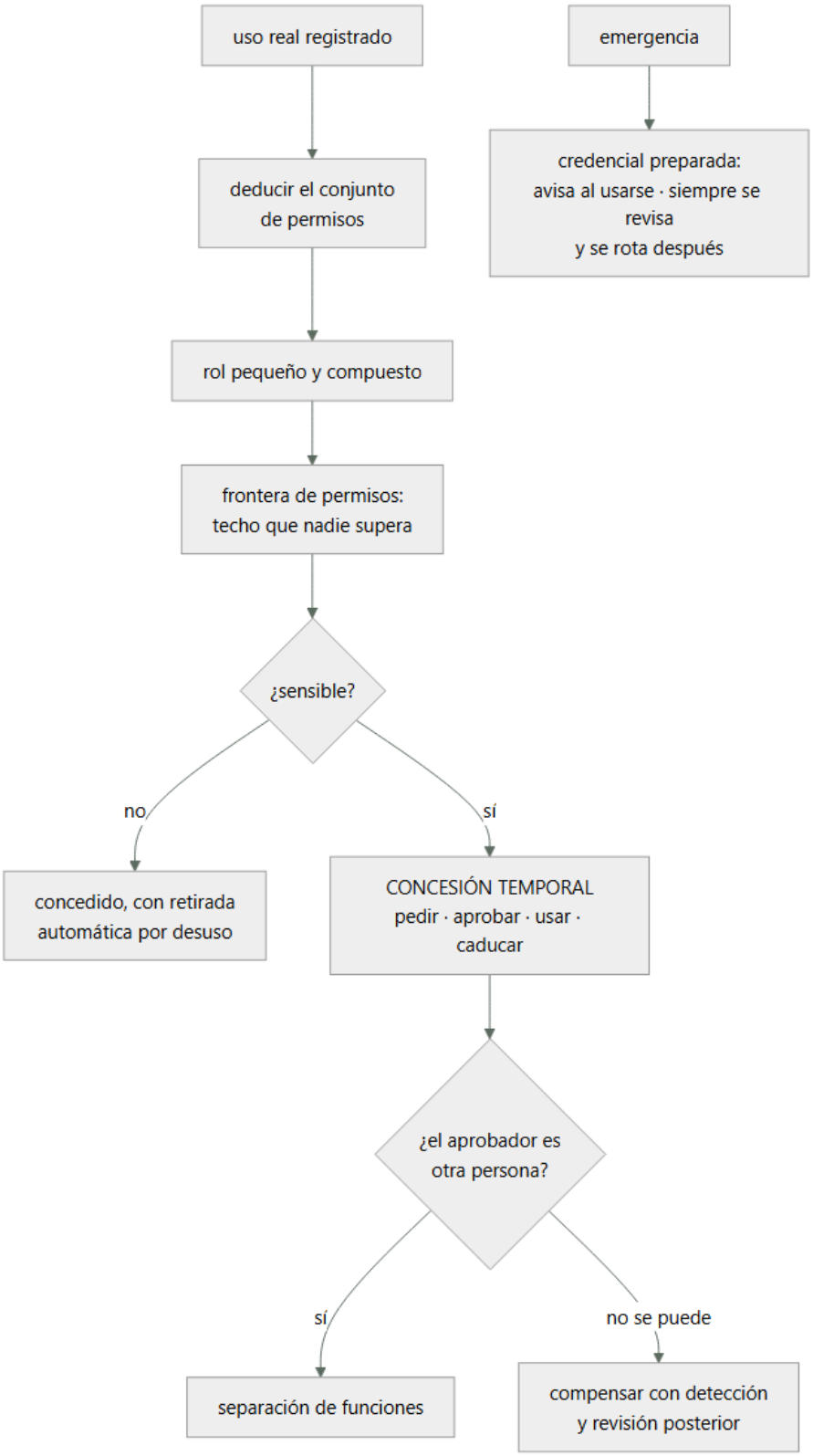
Concepto	Comprensión verificable
privilegio mínimo	Cada identidad puede hacer solo lo que necesita. Es un proceso continuo: sin algo que los retire, los permisos solo crecen.
acceso permanente	Permiso disponible sin pedirlo. Es el que un atacante encuentra ya concedido; el objetivo es que no exista para lo sensible.
concesión temporal	Permiso que se pide, se aprueba, dura un rato y desaparece solo. Debe ser más rápido que cualquier atajo.
separación de funciones	Que ninguna persona pueda completar sola una acción dañina de principio a fin.
acceso de emergencia	Credencial preparada para cuando todo lo demás falla. Debe existir, avisar al usarse y revisarse siempre.
frontera de permisos	Límite superior que ninguna concesión puede superar, aunque quien la conceda tenga más permisos.

Modelo mental

Gobernar no significa aprobar cada cambio: significa codificar límites, evidencia y responsabilidades para que los equipos puedan avanzar con seguridad.

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. Deducir los permisos del uso

El privilegio mínimo falla casi siempre por la misma razón: nadie sabe qué permisos hacen falta, así que se conceden de más «por si acaso» y nunca se quitan.

```
se concede al empezar el proyecto, con margen
se añade cada vez que algo falla por permisos
```

```
no se quita nunca, porque quitar puede romper algo
→ el conjunto solo crece
```

Y la técnica que lo resuelve no es adivinar mejor: es mirar lo que se ha usado de verdad. Los tres proveedores registran cada llamada con su identidad, y de ahí sale el conjunto real:

```
1. conceder amplio, en un entorno inferior, durante 30-60 días
2. leer el registro de auditoría: qué acciones se ejecutaron,
   sobre qué recursos, con qué identidad
3. construir el permiso a partir de esa lista
4. aplicarlo en modo de aviso antes de bloquear
5. y volver a revisar a los tres meses
```

Y el paso 4 es el que hace viable el ejercicio: primero avisar de lo que se habría bloqueado, y solo bloquear cuando la lista de avisos está vacía. Es la misma secuencia de adopción que este programa lleva usando desde la clase 091.

Y dos cautelas honestas sobre el método:

```
lo que ocurre una vez al año no aparece en 60 días
  → cierres contables, restauraciones, migraciones
  → hay que preguntarlo, no solo observarlo
lo observado incluye lo que se hizo mal
  → si alguien leyó una tabla que no debía, aparecerá como «necesario»
```

Y el diseño de los roles, con sus antipatrones habituales:

```
mal   un rol por puesto: «desarrollador», que crece sin fin
mal   comodines: acciones y recursos abiertos «para simplificar»
mal   copiar el rol de alguien que ya trabajaba aquí

bien  conjuntos pequeños por capacidad concreta
      «leer registros de este servicio», «reiniciar estos despliegues»
      y se componen los que hagan falta
```

La frontera de permisos es la pieza que evita que todo esto se deshaga por un descuido:

```
aunque alguien conceda un permiso amplio, la frontera lo recorta
→ «ninguna identidad de este proyecto puede borrar copias de seguridad»
→ «ninguna carga puede modificar el registro de auditoría»
```

Y es lo que permite delegar la concesión sin delegar el riesgo: los equipos se dan permisos entre ellos, y hay cosas que nadie puede concederse.

2. Que caduquen por defecto

La distinción que cambia el resultado:

```
no es «tiene demasiados permisos»
es «los tiene SIEMPRE»
```

Un atacante que roba una credencial encuentra lo que estaba concedido en ese momento. Si lo sensible no está concedido casi nunca, el robo vale mucho menos.

```
administrador permanente        el atacante es administrador
administrador bajo petición     el atacante tiene lo básico
                                y pedir más deja rastro y avisa
```

Y el ciclo de una concesión temporal:

```
PEDIR      con motivo y duración; en segundos, no en un formulario largo
APROBAR    otra persona, o automático según reglas para lo de menor riesgo
USAR       con todo registrado y asociado a la petición
CADUCAR    solo, sin que nadie tenga que acordarse
```

Y el requisito que decide si se adopta, que viene de la clase 133:

```
pedir acceso debe ser MÁS RÁPIDO que cualquier atajo
→ si tarda tres días, la gente conservará accesos permanentes
  «por si acaso», y tendrá razón
```

Y qué se aprueba automáticamente y qué no:

```
automático   lectura en entornos inferiores, diagnóstico, registros
             → con registro y caducidad corta
revisión     escritura en producción, datos personales, claves,
             permisos sobre permisos
```

Y una decisión que suele olvidarse: qué pasa durante un incidente. Si la concesión temporal depende de un sistema que puede estar caído, hay que tener el acceso de emergencia del apartado cuarto, o el proceso se rodeará justo cuando importa.

Las identidades de máquina, que son la mayoría y a las que casi nunca se aplica esto:

```
en una organización mediana hay 5-20 identidades de carga por persona
y casi todas tienen permisos permanentes
```

Y lo que sí se puede aplicar a ellas:

```
permisos deducidos del uso, igual que a las personas
credenciales de corta vida por federación            clase 098
retirada automática de lo que no se usa
y una identidad por carga, no una compartida por diez
```

La última es la que más cuesta y más rinde: una identidad compartida hace imposible saber quién hizo qué y obliga a que sus permisos sean la unión de todos los usos.

3. Separación de funciones y acceso de emergencia

La separación de funciones existe para que ninguna persona pueda completar sola una acción dañina.

```
quien escribe el código          ≠  quien lo aprueba          clase 097
quien pide un acceso             ≠  quien lo concede
quien despliega en producción    ≠  quien puede borrar el registro
quien administra las claves      ≠  quien administra los datos cifrados
quien crea un proveedor          ≠  quien aprueba sus pagos
```

Y una que merece destacarse porque suele estar mal: nadie debe poder modificar el registro de auditoría, ni siquiera quien administra. Se consigue con la inmutabilidad de la clase 112 y con una cuenta separada.

Y el límite honesto en equipos pequeños:

```
con cuatro personas no se pueden separar seis funciones
→ lo que se hace entonces:
   registrar todo lo que una sola persona pudo hacer sola
   revisarlo después, en una rutina fija
   y avisar automáticamente de las combinaciones peligrosas
→ es un control compensatorio, y hay que llamarlo así
```

El acceso de emergencia tiene que existir. Si no existe, se improvisa, y lo improvisado no tiene control ninguno.

```
preparado de antemano, no creado durante el incidente
fuera de las dependencias que pueden estar caídas
  → no depende del sistema de identidad corporativo si ese puede fallar
guardado de forma que usarlo requiera un acto deliberado
con aviso automático a varias personas al usarse
con revisión OBLIGATORIA de cada uso, siempre
y rotado después de usarse
```

Y las dos cifras que dicen si está bien:

```
usos al año                pocos; si son muchos, falta un camino normal
usos sin revisión posterior  cero, sin excepciones
```

Y un ensayo que hay que hacer, porque es exactamente el tipo de cosa que se pudre: usarlo a propósito una vez al año y comprobar que la credencial funciona, que el aviso llega y que la revisión ocurre. Es la clase 131 aplicada aquí.

4. Retirar sin depender de nadie

La práctica habitual es la revisión periódica de accesos: cada trimestre alguien recibe una lista y confirma quién debe conservar qué. Y funciona mal por un motivo predecible:

```
la lista tiene 400 líneas
quien revisa no sabe para qué necesita cada persona cada permiso
quitar algo puede romper el trabajo de alguien
→ se aprueba todo
```

Es la ley 15 y la ley 16 a la vez, y produce una firma que dice que se revisó.

Lo que sí funciona:

```
RETIRADA AUTOMÁTICA POR DESUSO
  «este permiso no se ha usado en 90 días → se retira»
  con aviso previo y forma de recuperarlo en segundos
```

```
→ no requiere que nadie juzgue nada
→ y el error se corrige solo: quien lo necesite lo vuelve a pedir
```

Y el requisito que lo hace aceptable es el mismo de siempre: recuperar el permiso tiene que ser inmediato. Con eso, retirar de más deja de dar miedo.

Y lo que sí merece revisión humana, que es mucho menos:

```
quién tiene permisos sobre permisos
quién puede llegar a datos personales
qué identidades tienen acceso permanente a producción
las excepciones vivas y su motivo
```

Cuatro listas cortas se revisan de verdad; una lista de cuatrocientas líneas, no.

Y lo que hay que vigilar de forma continua:

```
permisos concedidos y no usados nunca
identidades con acceso permanente a lo sensible
concesiones temporales: cuántas, de qué duración, aprobadas por quién
usos del acceso de emergencia
identidades compartidas
credenciales de larga duración que quedan
combinaciones que rompen la separación de funciones
```

Y la lista de comprobación de la clase:

- los permisos se deducen del uso registrado, no de suposiciones
- se aplica en modo aviso antes de bloquear
- se pregunta por lo que ocurre una vez al año y no aparece en el registro
- los roles son conjuntos pequeños compuestos, sin comodines
- hay fronteras que ninguna concesión puede superar
- lo sensible no está concedido de forma permanente
- pedir acceso temporal tarda menos que cualquier atajo
- nadie puede modificar el registro de auditoría
- las funciones críticas están separadas, o hay control compensatorio escrito
- existe acceso de emergencia preparado, con aviso y revisión obligatoria
- se ensaya el acceso de emergencia una vez al año
- la retirada por desuso es automática y recuperar es inmediato
- las identidades de carga tienen el mismo tratamiento que las personas

Y el cierre que enlaza con la clase siguiente: con los permisos acotados, queda la otra dimensión que la clase 133 enumeró como capa independiente —desde dónde se puede alcanzar cada cosa— y con ella el tráfico que sale, que es por donde se van los datos. Es la materia de la clase 135.

Ejemplo trabajado

CloudShop aplica el privilegio mínimo empezando por medir. El ejercicio produce tres números que orientan el resto: cuántos permisos concedidos no se usan, cuántas identidades tienen acceso permanente a producción y cuánto se tarda en pedir un acceso.

La medición inicial.

```
identidades humanas                                    41
identidades de carga                                  312
relación                                              7,6 por persona

permisos concedidos (acciones distintas por identidad)  8.940
usados al menos una vez en 90 días                    1.310
sin usar nunca                                        7.630   (85 %)

identidades con acceso permanente a producción           34
de ellas, humanas                                        19
identidades compartidas                                   6
tiempo medio para conseguir un acceso nuevo           3 días
```

Ochenta y cinco por ciento de los permisos concedidos no se ha usado nunca.

La deducción desde el uso.

Se tomó el registro de auditoría de sesenta días para las diez identidades de carga más importantes:

```
servicio de pedidos
  permisos concedidos                                  412
```

```
  acciones distintas ejecutadas                          23
  recursos distintos tocados                              7
  permiso deducido                      23 acciones sobre 7 recursos
```

Y el modo de aviso antes de bloquear, durante tres semanas:

```
semana 1   avisos de lo que se habría bloqueado                41
           de ellos, legítimos y no observados en los 60 días   9
             → 6 eran del cierre mensual
             → 3 eran de un procedimiento de recuperación
semana 2   avisos                                               7
semana 3   avisos                                               0
→ se activó el bloqueo
```

Los nueve del primer punto confirman la cautela del apartado primero: lo que ocurre una vez al mes no está en el registro de sesenta días, y hubo que preguntarlo.

```
permisos del servicio de pedidos            412            29
las 10 identidades principales            3.180           186
incidentes causados por falta de permisos     —          2 en 6 meses
  → los 2, procedimientos anuales; añadidos
```

Las concesiones temporales.

El acceso permanente a producción se retiró y se sustituyó por concesión temporal.

```
identidades humanas con acceso
permanente a producción                     19             0
tiempo para pedir acceso                  3 días          90 s
aprobación                              por correo    otra persona del
                                        y a mano       equipo dueño (095),
                                                       o automática si es
                                                       lectura
duración por defecto                        —             2 h
```

Y el uso real en seis meses:

```
concesiones solicitadas                                   418
aprobadas automáticamente (lectura, entornos inferiores)  291
aprobadas por otra persona                                121
rechazadas                                                  6
tiempo medio hasta obtenerla                               90 s
duración media usada                                      34 min
```

Y el efecto sobre el ejercicio de la clase 133:

```
alcance desde el portátil de desarrollo
  antes    acceso total, con claves en disco
  después  nada sin una concesión aprobada y registrada
```

La revisión trimestral que era teatro.

```
últimas cuatro revisiones antes del cambio
  líneas a revisar por trimestre                ~400
  tiempo dedicado por revisor                   ~20 min
  permisos retirados en las cuatro revisiones      3
  proporción aprobada sin cambios               99,3 %
```

Tres permisos retirados en un año. Se sustituyó por retirada automática:

```
permisos retirados en 6 meses                1                6.840
reclamaciones por retirada indebida          —                   31
de ellas, recuperadas en                     —                < 2 min
tiempo humano invertido                  80 min/trim             0
```

Treinta y una retiradas fueron molestas y se recuperaron en menos de dos minutos, que es exactamente lo que hace aceptable el mecanismo.

Y las cuatro listas cortas que sí se revisan a mano:

```
quién puede conceder permisos                 6 personas
quién puede llegar a datos personales         4 identidades
acceso permanente a producción                0 humanas, 11 cargas
excepciones vivas                            31, todas con fecha
```

Las identidades compartidas.

```
6 identidades compartidas, usadas por 23 personas
```

```
efecto  imposible saber quién hizo qué
        y sus permisos eran la unión de todos los usos: 1.140 acciones

tras separarlas en 23 identidades individuales
  permisos medios por identidad                              41
  acciones que nadie podía atribuir                    de 100 % a 0 %
```

El acceso de emergencia, y sus tres usos.

```
creado: credencial sellada, fuera del sistema de identidad corporativo
avisa a 4 personas al usarse
revisión obligatoria de cada uso

usos en 12 meses                                              3
  1. caída del proveedor de identidad                    legítimo
  2. incidente con el sistema de concesiones caído       legítimo
  3. alguien tenía prisa y no quiso esperar 90 s         NO legítimo
     → se habló, y se revisó por qué le pareció más rápido

usos sin revisión posterior                                   0
ensayo anual                                          ejecutado
  hallazgo del ensayo   el aviso llegaba a un buzón de un equipo
                        disuelto (el mismo problema de la clase 131)
```

La separación de funciones, con un equipo pequeño.

```
funciones que se pudieron separar                             5
funciones que NO, por tamaño del equipo                       2
  administrar claves y administrar datos cifrados
  aprobar un proveedor y aprobar su pago
control compensatorio escrito                              sí
  registro de las acciones combinadas
  revisión mensual de 20 min
  alerta automática cuando la misma persona hace ambas
alertas disparadas en 6 meses                                 4
  de ellas, legítimas                                         4
```

A los seis meses.

```
permisos concedidos                          8.940          2.100
sin usar nunca                               85 %            9 %
identidades humanas con acceso permanente
a producción                                   19              0
identidades compartidas                         6              0
tiempo para conseguir acceso                3 días           90 s
fronteras de permisos definidas                 0              4
permisos retirados en 6 meses                   1          6.840
recuperación tras retirada indebida             —          < 2 min
tiempo humano en revisiones                80 min/trim     20 min/mes
usos del acceso de emergencia sin revisar       —              0
acciones no atribuibles a una persona         100 %            0 %
```

La lección que esta clase traslada a la parte 11: el 85 % de los permisos concedidos no se había usado nunca, y la revisión trimestral que existía para detectarlo retiró tres en un año. Lo que sí funcionó fue quitar el juicio humano de la ecuación: retirar automáticamente por desuso y hacer que recuperar cueste dos minutos. Y el cambio que hizo posible eliminar diecinueve accesos permanentes a producción no fue una política: fue bajar de tres días a noventa segundos el tiempo de pedir uno temporal, que es la ley 16 aplicada en la dirección correcta.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-11-security-governance-finops/134-minimo-privilegio-acceso-temporal-y-separacion-de-funciones/lab.py
```

El laboratorio selecciona el motor de práctica iam 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 revision-accesos en evidence/ usando la plantilla indicada.
5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

El artefacto principal es una matriz de acceso mínimo con prueba de denegación. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye revision-accesos 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
Los permisos solo crecen y nadie se atreve a quitar ninguno	Se conceden por suposición y no hay ningún mecanismo que los retire	Deduce el conjunto del registro de uso, aplica en modo aviso antes de bloquear y añade retirada automática por desuso.
La revisión trimestral de accesos se aprueba entera sin cambios	Cuatrocientas líneas que nadie puede juzgar; leyes 15 y 16	Retira automáticamente lo no usado y deja para revisión humana solo cuatro listas cortas y sensibles.
Robar una credencial da acceso administrativo inmediato	El acceso sensible es permanente	Conviértelo en concesión temporal con caducidad, y haz que pedirla tarde menos que cualquier atajo.
No se puede saber quién ejecutó una acción	Hay identidades compartidas, cuyos permisos son además la unión de todos los usos	Una identidad por persona y por carga; separa las compartidas aunque cueste.
Durante un incidente nadie puede acceder porque el sistema de concesiones está caído	No hay acceso de emergencia, o depende de lo mismo que falló	Prepara una credencial sellada fuera de esas dependencias, con aviso automático, revisión obligatoria de cada uso y ensayo anual.
Un permiso concedido por error da acceso a algo crítico	No hay techo: quien concede puede conceder cualquier cosa	Define fronteras de permisos que ninguna concesión pueda superar, como borrar copias o tocar el registro de auditoría.

■ 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 deduce un conjunto de permisos a partir del uso y qué se escapa a ese método?

2. ¿Por qué el problema es tener permisos siempre y no tenerlos de más?
3. ¿Qué hace falta para que una concesión temporal se adopte de verdad?
4. ¿Qué se hace cuando el equipo es demasiado pequeño para separar funciones?
5. ¿Por qué la retirada automática por desuso funciona mejor que la revisión periódica?

Referencias

- AWS (2025). IAM Access Analyzer: policy generation from access activity — deducir permisos del uso registrado. <https://docs.aws.amazon.com/IAM/latest/UserGuide/access-analyzer-policy-generation.html>
- Microsoft (2025). Privileged Identity Management: just-in-time access — concesión temporal, aprobación y caducidad. <https://learn.microsoft.com/entra/id-governance/privileged-identity-management/pim-configure>
- Google Cloud (2025). Policy Intelligence and recommender — permisos concedidos y no usados. <https://cloud.google.com/policy-intelligence/docs/role-recommendations-overview>
- NIST (2020). SP 800-53: AC-5 separation of duties, AC-6 least privilege — definición y controles compensatorios. <https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final>
- AWS (2025). Permissions boundaries — techo que ninguna concesión delegada puede superar. <https://docs.aws.amazon.com/IAM/latest/UserGuide/accesspoliciesboundaries.html>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 11 en PDF · Manual integral

Anterior	Índice	Siguiente
← 133 · Zero Trust y defensa en profundidad	Parte 11 · Programa	135 · Segmentación, perímetro, WAF, DDoS y egress →

Evaluación — 134 Mínimo privilegio, acceso temporal y separación de funciones

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 revision-accesos 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.

135 — Segmentación, perímetro, WAF, DDoS y egress

Parte: 11 — Seguridad, gobierno, cumplimiento y FinOps Nivel: avanzado · Horas estimadas: 4 Laboratorio: network · Estado: EXECUTABLECORE

Propósito

Usar la red como la capa independiente que la clase 133 pedía: no como el control que decide quién entra, sino como el que limita hasta dónde se llega y qué puede salir. La clase defiende que la mitad olvidada es la salida —los datos se van por ahí, no por la entrada—, muestra por qué segmentar por identidad escala mejor que por dirección en entornos dinámicos, y trata los servicios de perímetro con honestidad: qué detienen de verdad, qué no, y por qué acaban en modo aviso durante catorce meses.

Resultados de aprendizaje

Al finalizar podrás:

1. Segmentar con denegación por defecto y una secuencia de adopción viable.
2. Controlar la salida, que es por donde se van los datos.
3. Situar el filtro de aplicación y la protección de denegación de servicio por lo que hacen de verdad.
4. Usar conectividad privada para que el tráfico no salga a la red pública.
5. Registrar lo suficiente para investigar sin arruinarse.

Conceptos centrales

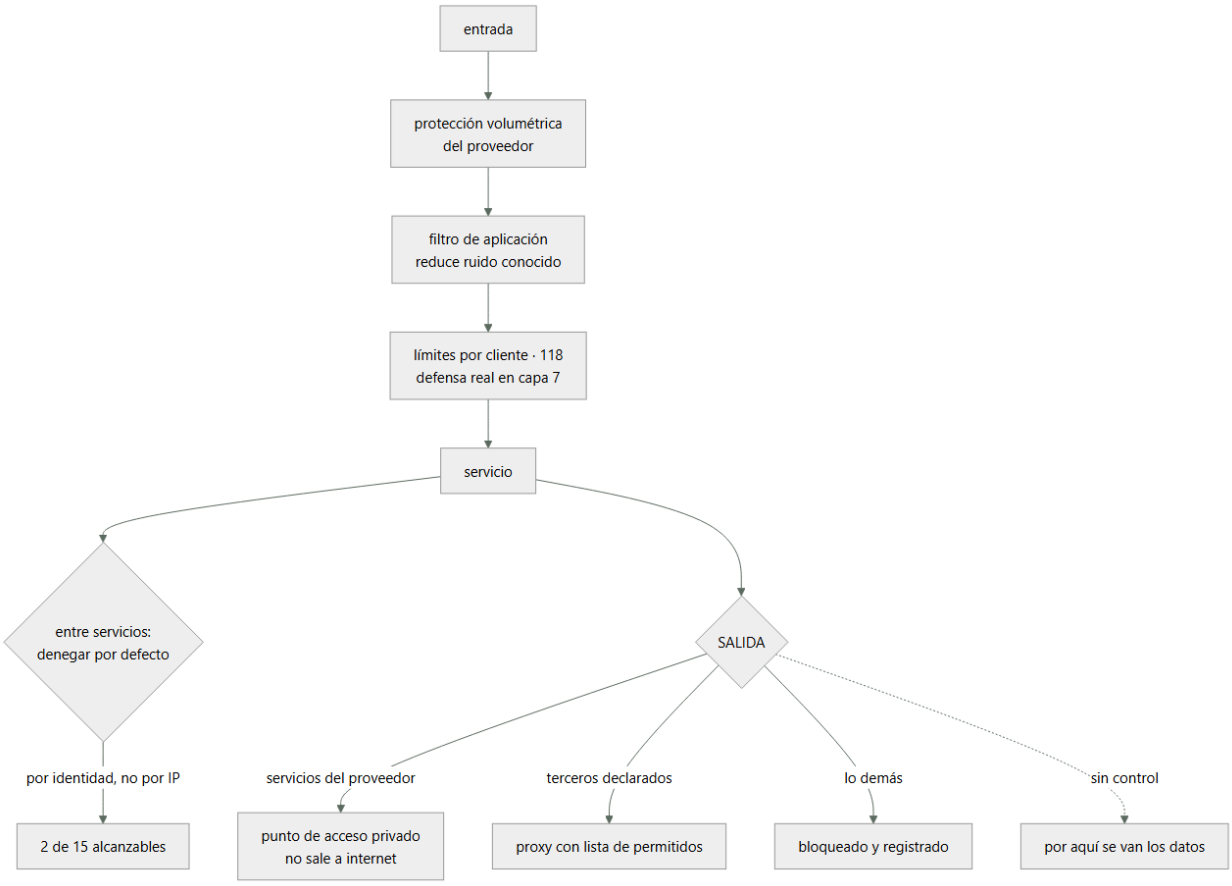
Concepto	Comprensión verificable
denegación por defecto	Solo se permite lo declarado. Es lo que convierte la red en una capa; con permitir por defecto no limita nada.
segmentación por identidad	Autorizar por quién es la carga, no por su dirección. En entornos dinámicos, la dirección no identifica nada.
control de salida	Restringir a qué destinos externos puede conectar una carga. Es el control que limita la extracción de datos.
punto de acceso privado	Acceso a un servicio gestionado sin salir a la red pública, y restringido a tus propias cuentas.
filtro de aplicación	Inspecciona peticiones y bloquea patrones conocidos. Reduce ruido y no sustituye a corregir la vulnerabilidad.
registro de flujos	Anotación de las conexiones observadas. Es la base para pasar de permitir a denegar sin romper nada.

Modelo mental

Gobernar no significa aprobar cada cambio: significa codificar límites, evidencia y responsabilidades para que los equipos puedan avanzar con seguridad.

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. Segmentar, y cómo llegar a denegar por defecto

Tras la clase 133, la red ya no decide quién es quién. Lo que sigue haciendo, y muy bien, es limitar el alcance:

```
si un servicio comprometido puede alcanzar 14 de 15, el atacante
tiene 14 objetivos
si puede alcanzar 2, tiene 2
```

Y las unidades de segmentación, de más fuerte a más fina:

```
CUENTA O PROYECTO      la frontera más difícil de cruzar        clase 133
RED VIRTUAL            separa entornos y dominios
SUBRED                 separa capas: pública, privada, datos
GRUPO O POLÍTICA       entre cargas concretas
MALLA CON IDENTIDAD    autoriza por identidad de la carga, con cifrado
```

Y la elección entre las dos últimas es la decisión práctica:

```
POR DIRECCIÓN     funciona con recursos estables
                  y en contenedores las direcciones cambian a cada rato
                  → las reglas envejecen mal y acaban siendo amplias

POR IDENTIDAD     «el servicio pedidos puede llamar a precios»
                  → sobrevive a que todo se recree
                  → y añade autenticación mutua, que es una capa más
```

Y la secuencia de adopción, que es la misma que este programa lleva usando desde la clase 091 y que aquí es imprescindible:

```
1. OBSERVAR   activar el registro de flujos y construir el mapa real
              de quién habla con quién
2. AVISAR     poner las reglas en modo aviso; recoger lo que se habría
              bloqueado
3. DENEGAR    cuando la lista de avisos esté vacía durante dos semanas
```

Y lo que aparece en el paso 1 es siempre lo mismo y siempre sorprende: conexiones que nadie sabía que existían, igual que las dieciocho dependencias de la clase 124.

Y una advertencia sobre las reglas amplias: 0.0.0.0/0 en un grupo interno no es un error de configuración, es la forma que toma la ley 16 cuando denegar cuesta demasiado. Si aparece, la pregunta es por qué fue más fácil eso que declarar lo necesario.

2. La salida, que es por donde se van los datos

Casi todo el esfuerzo se dedica a quién entra. Y los datos salen:

```
un atacante que ya está dentro necesita sacar la información
un programa malicioso necesita hablar con quien lo controla
una biblioteca comprometida necesita enviar lo que recoge
```

Y el control de salida es lo que lo impide o lo hace evidente. Lo que se gana, en orden:

```
limitar la extracción de datos
cortar la comunicación con la infraestructura del atacante
y, de regalo, un inventario exacto de dependencias externas
```

La tercera es útil desde el primer día aunque no se bloquee nada: saber a cuántos destinos habla cada carga. La respuesta suele estar entre cinco y quince, y encontrar cuarenta es una señal.

Y las tres formas de controlarlo, que se combinan:

```
PUNTOS DE ACCESO PRIVADOS para los servicios del proveedor
  el tráfico no sale a la red pública
  y se restringe a tus propias cuentas
  → esto es lo que impide que alguien copie datos a un almacenamiento
    de OTRA organización usando el mismo servicio

PROXY CON LISTA DE PERMITIDOS para terceros
  por nombre de dominio, no por dirección
  → porque los servicios externos cambian de dirección constantemente

DENEGAR EL RESTO, y registrarlo
```

Y la parte incómoda: las listas por dominio son laboriosas. Un proveedor externo puede usar decenas de dominios y redes de distribución compartidas. Lo que hace viable el ejercicio:

```
empezar por las cargas que tocan datos sensibles, no por todas
aceptar listas más amplias para lo que no los toca
y medir: destinos por carga, y cuántos son desconocidos
```

Y un detalle que se olvida y que anula el control: la resolución de nombres. Si una carga puede consultar cualquier servidor de nombres, puede sacar datos por ahí. Obligar a usar el resolutor propio y registrar las consultas es barato y es de los registros más útiles que existen para detectar algo comprometido.

Y el efecto colateral que enlaza con la clase 142: la salida cuesta dinero. Las pasarelas de traducción de direcciones, el tráfico entre zonas y la transferencia hacia fuera suelen estar entre las tres mayores partidas de red, y casi nadie las mira.

3. Perímetro: qué detiene de verdad

El filtro de aplicación inspecciona peticiones y bloquea patrones conocidos.

```
detiene bien   exploración automatizada, intentos con patrones conocidos,
               ruido masivo, y da tiempo mientras se corrige algo
detiene mal    fallos de lógica de negocio, autorización rota,
               abuso con peticiones perfectamente válidas
no sustituye   a corregir la vulnerabilidad
```

Y su problema característico es el de siempre:

```
en modo bloqueo produce falsos positivos
→ un cliente legítimo deja de funcionar
→ se pasa a modo aviso «temporalmente»
→ y sigue en modo aviso catorce meses después
```

Es la ley 16 y la ley 15 juntas, exactamente igual que los escáneres de la clase 101, y se corrige igual:

```
activar por reglas, no todo de golpe
empezar por las que no dan falsos positivos
```

```
medir bloqueos legítimos frente a ilegítimos
y tener excepción por regla, con motivo y caducidad
```

La denegación de servicio tiene dos formas muy distintas:

```
VOLUMÉTRICA     saturar el enlace con tráfico
  → no se puede absorber por cuenta propia: hace falta el proveedor
  → es de las pocas cosas que se resuelven contratando

DE APLICACIÓN   peticiones válidas y caras, en cantidad
  → parece tráfico normal
  → aquí no hay producto que la resuelva sola: se defiende con
    límites por cliente (clase 118), descarte por sobrecarga (clase 130)
    y capacidad conocida (clase 129)
```

Y lo que decide el resultado en la segunda no es el filtro, es poder distinguir a los clientes: si todo el tráfico es anónimo e indistinguible, cualquier límite castiga también a quien no molesta.

Y una precaución de coste, que sorprende: absorber un ataque con autoescalado es pagarlo. Conviene tener un techo, exactamente por el mismo motivo que la clase 117 lo puso a las funciones.

La conectividad privada merece una nota aparte porque hace dos cosas a la vez:

```
el tráfico no atraviesa la red pública
y la política puede exigir que el servicio solo acepte peticiones
desde tus cuentas
→ lo segundo es un control de extracción de datos, no de red
```

La segunda línea es la que más aporta y la que menos se configura.

4. Ver lo que pasa por la red

Sin registro no se puede pasar de permitir a denegar, ni investigar nada después.

```
REGISTRO DE FLUJOS      quién habló con quién, cuánto y si se permitió
  para qué   construir el mapa real, detectar conexiones raras,
             investigar un incidente
  cuidado    es voluminoso y caro; se muestrea, y se guarda agregado

REGISTRO DE NOMBRES     qué dominios se consultaron
  para qué   es la señal más útil para detectar algo comprometido
  cuidado    casi nadie lo activa, y cuesta poco

REGISTRO DEL FILTRO     qué se bloqueó y por qué regla
REGISTRO DEL PROXY      qué destinos externos se usaron
```

Y con la disciplina de coste de la clase 121: el registro de flujos completo de un sistema grande cuesta más que el resto de la telemetría junta. Se muestrea, se agrega y se conserva poco tiempo en detalle.

Y lo que conviene vigilar de forma continua:

```
destinos externos nuevos por carga            ← lo más informativo
conexiones denegadas, agrupadas por origen
volumen de salida por carga, y su tendencia   ← extracción de datos
consultas de nombres a dominios recién creados
reglas con rangos amplios
reglas que no han permitido nada en 90 días   ← candidatas a borrar
```

La tercera es la que detecta una extracción en curso, y solo funciona si hay una base con la que comparar: el volumen normal de salida de cada carga.

Y la última aplica a la red lo mismo que la clase 134 hizo con los permisos: una regla que no permite nada es una regla que sobra.

Y la lista de comprobación de la clase:

- los entornos están en cuentas o proyectos distintos
- hay denegación por defecto entre servicios, alcanzada por observar, avisar y denegar
- la autorización entre cargas es por identidad, no por dirección
- los servicios del proveedor se usan por punto de acceso privado
- esos puntos restringen el acceso a tus propias cuentas
- la salida a terceros pasa por una lista de permitidos
- la resolución de nombres está obligada al resolutor propio y registrada
- está medido a cuántos destinos habla cada carga

- el filtro de aplicación está en bloqueo por regla, no en aviso indefinido
- hay límites y descarte para el abuso en capa de aplicación
- el autoescalado tiene techo, para no pagar un ataque
- se vigilan destinos nuevos y volumen de salida por carga
- las reglas que no permiten nada se retiran

Y el cierre que enlaza con la clase siguiente: si el objetivo es que los datos no se puedan aprovechar aunque alguien llegue a ellos, hace falta la capa que la clase 133 puso sobre el dato y no sobre el camino. Cómo se cifra, quién controla las claves y qué protege de verdad cada opción es la materia de la clase 136.

Ejemplo trabajado

CloudShop aplica segmentación y control de salida. Lo que encuentra al observar antes de bloquear es más valioso que el bloqueo mismo, y una de las cosas que aparece lleva once meses en producción.

Paso 1: observar. El mapa real.

```
servicios                                                  15
conexiones declaradas en la documentación                  23
conexiones observadas en 30 días de registro de flujos     58
conexiones que nadie esperaba                              35
```

Y entre las treinta y cinco:

```
11  servicios que se llamaban entre sí por caminos antiguos
 9  herramientas de terceros con agentes instalados
 7  trabajos programados que nadie recordaba
 5  conexiones desde entornos inferiores hacia producción   ← grave
 3  conexiones a destinos externos no documentados
```

Las cinco de dev a producción son las que la clase 133 había dado por cortadas: existían por reglas antiguas que nadie había retirado.

Paso 2: avisar.

```
semana 1   conexiones que se habrían bloqueado          412
           de ellas, legítimas y no observadas           18
semana 2                                                 47
semana 3                                                  6
semana 4                                                  0
→ se activó la denegación
```

Paso 3: el resultado, medido con el ejercicio de la clase 133.

```
servicios alcanzables desde uno comprometido    14 de 15       2 de 15
conexiones permitidas                             todas          41
reglas con rango amplio                              9            0
conexiones de entornos inferiores a producción       5            0
```

Y la decisión de hacerlo por identidad en vez de por dirección:

```
reglas que hubo que mantener                  310               41
reglas rotas por un despliegue (6 meses)       23                0
autenticación mutua entre servicios            no               sí
esfuerzo de mantenimiento                    alto             bajo
```

Las veintitrés reglas rotas por despliegues eran el argumento: en un entorno donde todo se recrea, una regla por dirección caduca sola.

La salida: cuarenta y un destinos y uno que no debía estar.

Antes de bloquear nada, solo midiendo:

```
destinos externos distintos, todas las cargas              41
esperados según documentación                              12
desconocidos                                               29
```

Y al clasificar los veintinueve:

```
14  redes de distribución de contenido de dependencias legítimas
 8  telemetría de bibliotecas de terceros
 4  actualizaciones automáticas de agentes
 2  servicios que un equipo probó y dejó configurados
 1  un almacenamiento de objetos de OTRA organización
```

El último se investigó de inmediato:

```
origen        un proceso de exportación de informes
destino       un almacenamiento de un antiguo proveedor de análisis
volumen       1,2 GB al mes, durante 11 meses
contenido     informes agregados de ventas, sin datos personales
causa         el contrato terminó y el proceso siguió ejecutándose
quién lo sabía  nadie
```

Once meses enviando datos de negocio a una organización que ya no era proveedora. No fue un ataque: fue un proceso que nadie retiró, y ninguna de las diez clases de la parte 10 lo habría detectado porque el sistema funcionaba correctamente.

Y es la ley 13 en su forma más cara: lo que nadie paró siguió ejecutándose.

```
destinos externos permitidos                    sin límite       17
tráfico a destinos no declarados               1,2 GB/mes         0
registro de consultas de nombres                   no            sí
destinos nuevos detectados en 6 meses               —             6
  de ellos, legítimos y añadidos                    —             5
  de ellos, investigados como sospechosos           —             1
```

Los puntos de acceso privados.

```
tráfico a servicios del proveedor         por internet    por punto privado
restricción a cuentas propias                 no              sí
coste de salida por ese tráfico            310 €/mes         0 €
```

Y la restricción a cuentas propias bloqueó, en un ensayo de la clase 131, una copia de datos a un almacenamiento externo hecha con credenciales legítimas: el permiso lo autorizaba y la política de red lo impidió. Es exactamente una capa independiente funcionando.

El filtro de aplicación, catorce meses en modo aviso.

```
activado hacía                                        14 meses
modo                                                  aviso
motivo                     falsos positivos en la API de socios
reglas activas                                           340
bloqueos que se habrían producido, al día               12.400
de ellos, legítimos (clientes reales)                      ~90
```

Noventa clientes legítimos al día bloqueados era inaceptable, así que todo estaba en aviso, incluidas las 300 reglas que no daban ningún falso positivo.

```
reglas en bloqueo                              0 de 340       308 de 340
reglas en aviso, con excepción y fecha             —             32
falsos positivos al día                            —              2
peticiones maliciosas bloqueadas al día            0          11.900
```

El coste de red, que nadie miraba.

```
partidas de red, al mes
  pasarelas de traducción de direcciones                  890 €
  tráfico entre zonas                                     640 €
  transferencia hacia internet                            410 €
  registro de flujos completo                             720 €
                                                          ■■■■■■
                                                        2.660 €
comparado con el cómputo                                9.200 €
proporción                                                 29 %
```

Y las correcciones, que en parte son las mismas del control de salida:

```
puntos de acceso privados                    no             sí     −310 €
tráfico entre zonas evitable                 sí         reparto por zona
                                                                   −380 €
registro de flujos                        completo      muestreado 1:10
                                                        + agregado  −540 €
total de red                               2.660 €        1.430 €
```

A los seis meses.

```
servicios alcanzables desde uno comprometido    14 de 15       2 de 15
conexiones observadas no documentadas              35             0
conexiones de entornos inferiores a producción      5             0
destinos externos permitidos                  sin límite         17
tráfico a destinos no declarados             1,2 GB/mes           0
registro de consultas de nombres                  no             sí
```

```
reglas del filtro en bloqueo                  0 de 340      308 de 340
peticiones maliciosas bloqueadas al día           0          11.900
coste mensual de red                          2.660 €        1.430 €
```

La lección que esta clase traslada a la parte 11: el hallazgo más grave no lo produjo ningún bloqueo, sino mirar a dónde salía el tráfico antes de bloquear nada: un proceso llevaba once meses enviando informes de ventas a un proveedor con el que ya no había contrato. Y el filtro de aplicación llevaba catorce meses sin bloquear absolutamente nada porque treinta y dos reglas de trescientas cuarenta daban falsos positivos, que es la ley 16 en su forma más pura: un control se apagó entero por el 9 % que molestaba.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-11-security-governance-finops/135-segmentacion-perimetro-waf-ddos-y-egress/lab.py
```

El laboratorio selecciona el motor de práctica network 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 arquitectura-defensiva 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 topología con rutas, puertos y controles. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

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

■ Criterio de aceptación

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

Errores frecuentes

Síntoma	Causa probable	Corrección
Un servicio comprometido puede alcanzar casi todos los demás	Se permite por defecto entre cargas del mismo entorno	Observa con registro de flujos, pasa a modo aviso y activa denegación por defecto cuando la lista esté vacía.
Las reglas de red se rompen con cada despliegue	Autorizan por dirección en un entorno donde todo se recrea	Autoriza por identidad de la carga, con autenticación mutua.
Se descubren datos saliendo hacia un destino externo desconocido	No hay control ni inventario de salida	Mide a cuántos destinos habla cada carga, pon lista de permitidos por dominio y obliga y registra la resolución de nombres.
El filtro de aplicación lleva meses sin bloquear nada	Unas pocas reglas dan falsos positivos y se apagó el conjunto entero	Activa el bloqueo regla a regla y deja en aviso solo las conflictivas, con excepción, motivo y caducidad.
Un permiso legítimo permite copiar datos a otra organización	El servicio gestionado se usa por internet y sin restricción de cuentas	Usa puntos de acceso privados y exige en la política que solo acepten peticiones desde tus cuentas.
La factura de red es una de las mayores y nadie la mira	Pasarelas de traducción, tráfico entre zonas y registro de flujos completo	Puntos privados, reparto por zona y muestreo del registro de flujos con agregación.

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 secuencia permite llegar a denegación por defecto sin romper nada?
2. ¿Por qué la segmentación por identidad envejece mejor que la basada en direcciones?
3. ¿Qué tres cosas aporta controlar la salida, y cuál sirve desde el primer día sin bloquear nada?
4. ¿Qué detiene y qué no detiene un filtro de aplicación?
5. ¿Qué dos cosas distintas aporta un punto de acceso privado?

Referencias

- NIST (2016). SP 800-125B / segmentación y microsegmentación — unidades de aislamiento y su aplicación. <https://csrc.nist.gov/pubs/sp/800/125/b/final>
- Kubernetes (2025). Network policies — denegación por defecto y autorización entre cargas. <https://kubernetes.io/docs/concepts/services-networking/network-policies/>
- AWS (2025). VPC endpoints and endpoint policies — acceso privado y restricción a cuentas propias. <https://docs.aws.amazon.com/vpc/latest/privatelink/vpc-endpoints.html>
- OWASP (2025). Web application firewall: benefits and limitations — qué filtra y qué no. <https://owasp.org/www-community/WebApplicationFirewall>
- Cloudflare (2025). DDoS: volumetric and application-layer attacks — diferencias y defensas aplicables. <https://developers.cloudflare.com/ddos-protection/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 11 en PDF · Manual integral

Anterior	Índice	Siguiente
← 134 · Mínimo privilegio, acceso temporal y separación de funciones	Parte 11 · Programa	136 · Cifrado, KMS, HSM, rotación y envelope encryption →

Evaluación — 135 Segmentación, perímetro, WAF, DDoS y egress

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 arquitectura-defensiva 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.

136 — Cifrado, KMS, HSM, rotación y envelope encryption

Parte: 11 — Seguridad, gobierno, cumplimiento y FinOps Nivel: avanzado · Horas estimadas: 4 Laboratorio: security · Estado: EXECUTABLECORE

Propósito

Poner el cifrado en su sitio, que no es donde suele ponerse. La clase empieza por la pregunta que casi nunca se hace —¿de qué protege exactamente?— y llega a una conclusión incómoda: el cifrado en reposo, que es lo que todo el mundo enseña en una auditoría, no protege del escenario más probable, porque la aplicación comprometida puede descifrar. A partir de ahí desarrolla lo que sí ayuda: separar quién usa una clave de quién accede al dato, el cifrado por sobres y sus consecuencias, y los tres fallos operativos que convierten una clave en una caída o en una pérdida irreversible.

Resultados de aprendizaje

Al finalizar podrás:

1. Responder de qué protege cada tipo de cifrado y de qué no.
2. Separar el acceso a la clave del acceso al dato.
3. Explicar el cifrado por sobres y qué implica rotar la clave maestra.
4. Evitar los tres fallos operativos: caída por límite, indisponibilidad entre regiones y borrado irreversible.
5. Decidir qué rotar, con qué frecuencia y por qué.

Conceptos centrales

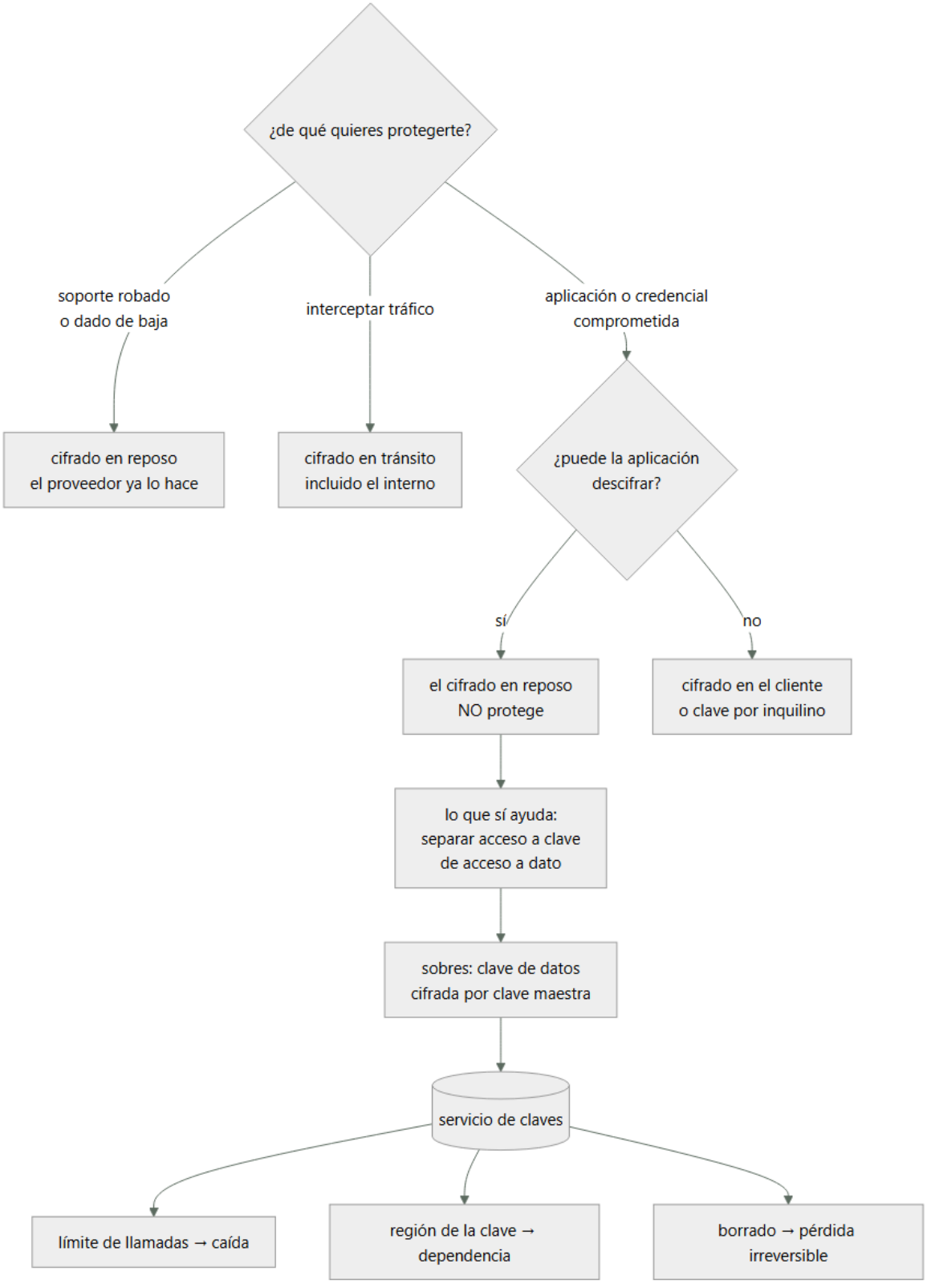
Concepto	Comprensión verificable
cifrado en reposo	Los datos se guardan cifrados en el soporte. Protege del acceso físico o directo al almacenamiento; no de una aplicación comprometida.
cifrado por sobres	Una clave de datos cifra el contenido y una clave maestra cifra la clave de datos. Solo la maestra vive en el servicio de claves.
separación de uso y administración	Quien puede cifrar y descifrar con una clave no es quien puede modificar su política ni borrarla.
rotación	Emitir una versión nueva de la clave. Con sobres, no vuelve a cifrar los datos existentes: acota cuánto se cifra con cada versión.
borrado de clave	Destruir la clave destruye los datos que protegía. Es irreversible y por eso hay periodo de espera.
cifrado en el cliente	El dato se cifra antes de llegar al servicio, que nunca ve el contenido. Es lo único que protege del propio servicio.

Modelo mental

Gobernar no significa aprobar cada cambio: significa codificar límites, evidencia y responsabilidades para que los equipos puedan avanzar con seguridad.

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. De qué protege cada cosa

La pregunta que ordena esta clase entera:

```
¿contra qué amenaza concreta protege este cifrado?
```

Y las tres amenazas habituales, con la respuesta honesta:

```
1. ALGUIEN SE LLEVA EL SOPORTE FÍSICO
   o el proveedor da de baja un disco sin borrarlo
   → cifrado en reposo. Resuelto.
   → y los tres grandes proveedores lo hacen por defecto
   → aportación adicional de configurarlo tú: pequeña

2. ALGUIEN INTERCEPTA EL TRÁFICO
   → cifrado en tránsito. Resuelto.
   → y el que suele faltar es el INTERNO: entre servicios,
     hacia la base de datos, hacia el caché

3. ALGUIEN COMPROMETE LA APLICACIÓN O UNA CREDENCIAL
   → el cifrado en reposo NO PROTEGE
   → porque la aplicación puede descifrar: para eso tiene la clave
```

Y la tercera es, con mucha diferencia, la más probable. De ahí la conclusión incómoda:

```
«ciframos en reposo» responde a una casilla de auditoría
y casi nada sobre el riesgo real
```

Lo que sí ayuda frente a la tercera, en orden de eficacia:

```
MENOS PERMISOS Y MENOS PERMANENTES        clase 134
  → el mejor control frente a este escenario no es criptográfico

SEPARAR ACCESO A LA CLAVE DEL ACCESO AL DATO
  → quien compromete el almacenamiento no obtiene la clave
  → es la independencia de capas de la clase 133

CLAVE DISTINTA POR INQUILINO O POR ÁMBITO
  → un compromiso alcanza una parte, no todo

CIFRADO EN EL CLIENTE
  → el servicio nunca ve el contenido
  → protege incluso del proveedor, y cuesta caro:
    no se puede buscar, ni indexar, ni ordenar por ese campo
```

Y la última tiene un ámbito realista: unos pocos campos, no la base entera. Números de tarjeta, documentos de identidad, datos de salud.

Y una lista que conviene tener presente porque son los casos donde el cifrado en reposo sí aporta de verdad:

```
copias de seguridad que se exportan
datos en almacenamiento de objetos con acceso amplio
discos que se clonan para depurar
instantáneas compartidas entre cuentas
```

En todos ellos el dato sale del contexto de la aplicación, y ahí la clave sí marca la diferencia.

2. Sobres, y qué significa rotar

Cifrar directamente con una clave del servicio gestionado no escala: cada operación sería una llamada remota. El patrón habitual es de dos niveles:

```
1. se genera una CLAVE DE DATOS aleatoria
2. esa clave cifra el contenido, localmente y rápido
3. la clave de datos se cifra con la CLAVE MAESTRA, en el servicio
4. se guarda el dato cifrado junto a la clave de datos cifrada
5. la clave de datos en claro se borra de memoria
```

Y para leer, se pide al servicio que descifre la clave de datos y se descifra el contenido en local.

Lo que se gana:

```
velocidad         el cifrado del contenido no pasa por la red
límites           una llamada al servicio por objeto, no por byte
alcance           una clave de datos por objeto, por fichero o por inquilino
```

Y la consecuencia que más se malinterpreta:

```
ROTAR LA CLAVE MAESTRA NO VUELVE A CIFRAR LOS DATOS

la versión nueva se usa para lo nuevo
lo viejo sigue cifrado con la versión anterior, que se conserva
→ rotar acota cuánto material se cifra con cada versión
→ NO invalida nada de lo anterior
```

Y de ahí la respuesta a «cada cuánto se rota», que depende de qué se persigue:

```
limitar el material por versión           rotación automática anual basta
responder a una sospecha de compromiso    rotar NO basta: hay que
                                          volver a cifrar y deshabilitar
                                          la versión antigua
cumplir una norma                         lo que diga la norma
```

Y una distinción práctica que evita mucho esfuerzo mal dirigido:

```
lo que se filtra y hay que rotar a menudo   credenciales, testigos,
                                            claves de API      clase 137
lo que casi nunca se filtra                 una clave maestra que no sale
                                            del servicio
```

Quién puede qué, que es la separación de la clase 134 aplicada aquí:

```
USAR       cifrar y descifrar con la clave
ADMINISTRAR  cambiar su política, deshabilitarla, borrarla
→ y no deben ser la misma identidad
→ ni la aplicación debe poder administrar la clave que usa
```

Y el resultado de hacerlo bien, medible: quien compromete la aplicación puede descifrar mientras tenga acceso, y no puede llevarse la clave ni conceder acceso a nadie más.

3. Tres fallos operativos que cuestan caro

1. El límite de llamadas al servicio de claves.

Es una causa real de caídas y sorprende a todo el mundo:

```
el servicio de claves tiene un límite de peticiones por segundo
si cada operación pide descifrar una clave de datos, se alcanza
→ y entonces falla todo lo que necesita descifrar, a la vez
```

Lo que lo evita:

```
cachear la clave de datos descifrada en memoria, con caducidad corta
una clave de datos por lote o por fichero, no por registro
y vigilar las llamadas por segundo como cualquier otra saturación
```

2. La clave vive en una región.

```
datos replicados en tres regiones
clave maestra en una sola
→ si esa región no responde, los datos de las otras dos
  no se pueden descifrar
```

Es una dependencia oculta que rompe el diseño de recuperación de la clase 088. Las salidas: claves multirregión, réplicas de clave, o una clave por región con una política que las mantenga equivalentes.

3. Borrar una clave borra los datos.

```
destruir la clave hace ilegible todo lo que protegía
es irreversible: la ley 14 en su forma más literal
```

Por eso los servicios imponen un periodo de espera —de días a un mes— entre solicitar el borrado y ejecutarlo. Y lo que hay que tener:

```
el borrado de claves en la frontera de permisos: que NADIE pueda
  hacerlo sin una aprobación aparte              clase 134
alerta inmediata cuando se solicita un borrado
y una comprobación antes de aprobarlo: ¿qué datos protege esta clave?
  → y esa pregunta debe tener respuesta escrita, no una investigación
```

La última exige llevar inventario de qué protege cada clave, que casi nadie tiene y que es lo primero que se echa en falta.

Y dos temas que conviene situar sin exagerarlos:

```
MÓDULO DEDICADO (dispositivo)
  hace falta cuando una norma lo exige o cuando la clave no debe poder
  salir jamás en ningún formato
  → para la mayoría, el servicio gestionado basta y es más fiable

APORTAR TU PROPIO MATERIAL DE CLAVE
  permite decir que el proveedor no generó la clave
```

```
  y sigue estando en su servicio para poder usarla
  → el beneficio real es de gobierno, no criptográfico
  → y añade la obligación de custodiar el material fuera
```

Y el coste, que también se decide aquí: el servicio de claves se factura por clave y por llamada, y un diseño con una clave por registro y sin caché es caro además de frágil.

4. Certificados y cifrado en tránsito

El cifrado en tránsito público está resuelto desde hace años. El que falta es el interno:

```
entre servicios dentro de la red              a menudo en claro
hacia la base de datos y el caché             a menudo en claro
entre zonas del mismo proveedor               depende del servicio
hacia servicios gestionados                   casi siempre cifrado
```

Y el argumento de «es red interna» es exactamente el modelo de perímetro que la clase 133 descartó. La forma sana de resolverlo sin gestionar certificados a mano es la de la clase 135: autenticación mutua gestionada por la malla, que cifra y autentica a la vez.

Los certificados son la causa clásica de caídas evitables:

```
caduca un certificado interno un domingo
nadie lo vigilaba porque «lo renovamos hace un año»
→ caída total, y el procedimiento de renovación no está probado
```

Lo que lo evita, y ya estaba en la clase 125 como indicador adelantado:

```
emisión y renovación automáticas, siempre
inventario de todos los certificados, incluidos los internos
  y los que están dentro de imágenes de contenedor
alerta con 30 y con 14 días de margen
y un ensayo: dejar caducar uno a propósito en preproducción
```

La segunda línea es donde aparecen las sorpresas: certificados incrustados en imágenes, en configuraciones de clientes y en dispositivos, que ningún inventario automático encuentra.

Y dos precauciones más:

```
la validación del certificado del servidor a veces está desactivada
  «porque daba problemas» → eso anula el cifrado como control
las cadenas de confianza internas caducan también
  y una autoridad interna caducada afecta a todo a la vez
```

Y la lista de comprobación de la clase:

- ■ está escrito de qué amenaza protege cada cifrado que se aplica
- ■ el tráfico interno entre servicios va cifrado
- ■ la validación de certificados no está desactivada en ningún cliente
- ■ quien usa una clave no puede administrarla
- ■ hay clave distinta por ámbito o por inquilino donde importa
- ■ los campos más sensibles se cifran en el cliente
- ■ hay caché de claves de datos y se vigilan las llamadas por segundo
- ■ las claves están disponibles en todas las regiones donde hay datos
- ■ borrar una clave requiere aprobación aparte y dispara alerta
- ■ existe inventario de qué protege cada clave
- ■ los certificados se renuevan solos y hay inventario, incluidos los internos
- ■ se ensaya la caducidad de un certificado en preproducción

Y el cierre que enlaza con la clase siguiente: las claves maestras no salen del servicio, y las credenciales que las cargas usan para todo lo demás sí circulan, se copian y se filtran. Cómo se guardan, cómo se entregan y cómo se rotan es la materia de la clase 137.

Ejemplo trabajado

CloudShop responde a una auditoría con «todo está cifrado en reposo y en tránsito». El ejercicio consiste en comprobar qué protege eso de verdad, y termina con una caída causada por el propio servicio de claves.

La revisión de la respuesta a la auditoría.

```
afirmación                    realidad comprobada
«cifrado en reposo»           sí, con claves gestionadas por el proveedor
                              en 14 de 14 almacenes
«cifrado en tránsito»         sí hacia fuera
```

```
                                  NO entre 11 de 15 servicios internos
                                  NO hacia la base de datos ni el caché
```

Y la pregunta del apartado primero, aplicada al escenario más probable:

```
si se compromete la credencial del servicio de pedidos,
¿qué impide leer la base entera?
  el cifrado en reposo              NO: la aplicación descifra
  los permisos                      sí, y son de lectura completa
  la red                            sí, tras la clase 135
  el cifrado por campo              no existía
```

El cifrado en reposo no aportaba nada frente al riesgo principal, y eso era exactamente lo que la respuesta a la auditoría daba a entender.

Lo que se hizo, en orden de eficacia.

```
1. cifrado interno entre servicios (autenticación mutua de la malla)
   11 de 15 servicios pasaron de claro a cifrado y autenticado
   coste: activar una opción; el mayor trabajo fue el inventario

2. separar uso de administración de las claves
   antes: la identidad de la aplicación podía modificar la política
          de su propia clave
   después: solo usar; administrar queda en una identidad aparte

3. clave por inquilino en el almacén de documentos de clientes
   antes: una clave para 190 clientes
   después: una por cliente
   → un compromiso alcanza los datos de uno, no de todos

4. cifrado en el cliente para 3 campos
   documento de identidad, cuenta bancaria y datos de salud
   → el servicio de datos nunca ve el contenido
   coste asumido: esos 3 campos no se pueden buscar ni ordenar
```

Y el efecto medido en el ejercicio de alcance de la clase 133:

```
datos legibles al comprometer pedidos      base completa    pedidos, sin
                                                            los 3 campos
datos legibles al comprometer el almacén
de documentos                              190 clientes      1 cliente
```

La caída por el límite de llamadas.

```
09:40  campaña; el tráfico se triplica
09:41  el servicio de documentos pide descifrar una clave de datos
       POR CADA lectura
09:41  llamadas al servicio de claves: de 400/s a 3.100/s
09:42  límite alcanzado: 2.500/s
09:42  errores en CUANTO necesitaba descifrar, a la vez
09:58  se mitiga subiendo el límite (petición al proveedor: 16 min)
duración                                                  18 min
```

Y el diagnóstico: una clave de datos por registro, sin caché.

```
clave de datos                          por registro     por documento
caché de claves descifradas             no               sí, 5 min
llamadas al servicio de claves en el pico  3.100/s          38/s
coste mensual del servicio de claves       410 €            12 €
límite alcanzado desde entonces             —               nunca
vigilancia de llamadas por segundo         no              sí, con alerta
```

Ochenta veces menos llamadas y un coste treinta veces menor, con el mismo nivel de protección.

La dependencia entre regiones, encontrada en un ensayo.

En un experimento de la clase 131 —pérdida de una región—:

```
datos replicados en 3 regiones                        sí
claves maestras                          todas en la región principal
resultado del ensayo   los datos de las otras dos regiones no se podían
                       descifrar con la principal caída
plan de recuperación   quedaba invalidado por completo
```

```
claves                                    1 región      multirregión (4)
                                                        o réplica por región
```

```
ensayo repetido                              falla        correcto
tiempo de recuperación con la región
principal caída                            imposible       11 min
```

El borrado que casi ocurre.

```
mes 7   limpieza de recursos antiguos por un guion automático
        marcó para borrar 6 claves «sin uso reciente»
        de ellas, 1 protegía las copias de seguridad de 2023-2024
detectado por   la alerta de solicitud de borrado, añadida 2 meses antes
periodo de espera del proveedor                        30 días
tiempo hasta cancelarlo                                 40 min
```

Y las dos correcciones:

```
borrado de claves                  permitido a la
                                   identidad de limpieza    en la frontera
                                                            de permisos
aprobación                              no              2 personas
inventario de qué protege cada clave    no              sí, 22 claves
alerta al solicitar borrado          añadida en el mes 5   sí
```

Y la fila del inventario es la que hizo posible responder en cuarenta minutos: sin ella, averiguar qué protegía esa clave habría llevado días.

Los certificados.

```
inventario inicial
  certificados públicos, renovados automáticamente         14
  certificados internos, renovados a mano                  31
  certificados dentro de imágenes de contenedor             6   ← nadie los
                                                                conocía
  clientes con validación desactivada                       4
  autoridad interna, caduca en                          14 meses

renovación automática                       14 de 51       51 de 51
validación desactivada                         4              0
alerta a 30 y 14 días                         no             sí
ensayo de caducidad en preproducción          no          semestral
caídas por certificado caducado            2 / 2 años         0
```

Y el ensayo del primer semestre encontró que el procedimiento de renovación de la autoridad interna no existía, con catorce meses de margen para escribirlo.

A los seis meses.

```
servicios con tráfico interno cifrado         4 de 15       15 de 15
identidades que administran su propia clave      11             0
claves por inquilino donde importa                1           190
campos cifrados en el cliente                     0             3
llamadas al servicio de claves en el pico     3.100/s          38/s
coste del servicio de claves                   410 €           12 €
claves disponibles en todas las regiones          no            sí
inventario de qué protege cada clave              no        22 claves
borrados de clave posibles sin aprobación         sí            no
certificados con renovación automática        14 de 51      51 de 51
caídas por cifrado o claves                 3 / 6 meses         0
```

La lección que esta clase traslada a la parte 11: la respuesta «todo está cifrado» era literalmente cierta y no protegía del escenario más probable, porque la aplicación comprometida tiene por definición permiso para descifrar. Lo que sí cambió el riesgo fue separar quién usa una clave de quién la administra y trocearla por inquilino. Y el propio servicio de claves causó una caída de dieciocho minutos y estuvo a punto de destruir dos años de copias: la criptografía no falló ni una vez; falló todo lo que la rodea.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-11-security-governance-finops/136-cifrado-kms-hsm-rotacion-y-envelope-encryption/
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 jerarquia-claves 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 jerarquia-claves 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 responde a una auditoría que todo está cifrado y el riesgo real no baja	El cifrado en reposo no protege de una aplicación o credencial comprometida, que es el escenario más probable	Escribe de qué amenaza protege cada cifrado; añade permisos mínimos, separación de clave y dato, claves por ámbito y cifrado en el cliente para lo más sensible.
El servicio de claves alcanza su límite y falla todo lo que descifra	Una clave de datos por registro y sin caché	Agrupa por documento o lote, cachea la clave descifrada con caducidad corta y vigila las llamadas por segundo.
Con una región caída no se pueden descifrar datos de las demás	Las claves maestras viven en una sola región	Claves multirregión o réplica por región, y comprueba el plan de recuperación con un ensayo.
Un proceso automático solicita borrar claves en uso	El borrado no está en la frontera de permisos y no hay inventario de qué protege cada clave	Prohíbe el borrado en la frontera, exige aprobación aparte, alerta al solicitarlo y mantén el inventario.
Se rota la clave maestra y se supone que los datos antiguos quedan protegidos	Con sobres, rotar no vuelve a cifrar nada	Si hay sospecha de compromiso, vuelve a cifrar y deshabilita la versión antigua; la rotación periódica solo acota material por versión.
Un certificado caduca y provoca una caída	Renovación manual, inventario incompleto y certificados escondidos en imágenes	Renovación automática, inventario completo, alertas a 30 y 14 días y ensayo de caducidad en preproducción.

■ 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 el cifrado en reposo y de qué no?
2. ¿Qué aporta el cifrado por sobres y qué NO hace la rotación de la clave maestra?
3. ¿Por qué quien usa una clave no debe poder administrarla?
4. ¿Cómo puede el servicio de claves causar una caída y cómo se evita?
5. ¿Por qué borrar una clave es irreversible y qué hace falta antes de aprobarlo?

Referencias

- AWS (2025). KMS: envelope encryption and data key caching — mecanismo, límites y caché. <https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html>
- Google Cloud (2025). Cloud KMS: key rotation and key destruction — qué implica rotar y el periodo de espera al destruir. <https://cloud.google.com/kms/docs/key-rotation>
- Azure (2025). Key Vault: keys, secrets and access policies — separación entre uso y administración. <https://learn.microsoft.com/azure/key-vault/general/security-features>
- NIST (2020). SP 800-57: key management recommendations — periodos de uso, rotación y custodia. <https://csrc.nist.gov/pubs/sp/800/57/pt1/r5/final>
- OWASP (2025). Transport layer security cheat sheet — cifrado interno y validación de certificados. <https://cheatsheetseries.owasp.org/cheatsheets/TransportLayerSecurityCheatSheet.html>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 11 en PDF · Manual integral

Anterior	Índice	Siguiente
← 135 · Segmentación, perímetro, WAF, DDoS y egress	Parte 11 · Programa	137 · Gestión de secretos y credenciales de workloads →

Evaluación — 136 Cifrado, KMS, HSM, rotación y envelope encryption

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 jerarquia-claves 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.

137 — Gestión de secretos y credenciales de workloads

Parte: 11 — Seguridad, gobierno, cumplimiento y FinOps Nivel: avanzado · Horas estimadas: 4 Laboratorio: security · Estado: EXECUTABLECORE

Propósito

Resolver el problema de las credenciales que las cargas necesitan para todo lo demás. La clase ordena las soluciones por cuánto eliminan el problema en vez de por cuánto lo gestionan, y defiende que casi todo el mundo empieza por la tercera —comprar un almacén— cuando las dos primeras hacen desaparecer categorías enteras de riesgo. Y se detiene en el detalle que explica por qué sobreviven las credenciales eternas: una variable de entorno no se puede rotar sin reiniciar, así que rotar se vuelve una operación disruptiva y se deja de hacer.

Resultados de aprendizaje

Al finalizar podrás:

1. Ordenar las opciones por cuánto eliminan el problema, no por cuánto lo administran.
2. Sustituir credenciales estáticas por identidad federada donde sea posible.
3. Elegir el mecanismo de entrega sabiendo cuál permite rotar en caliente.
4. Rotar sin provocar cortes, con el patrón de dos credenciales válidas.
5. Reducir el rastro por el que un secreto acaba en sitios donde no debería.

Conceptos centrales

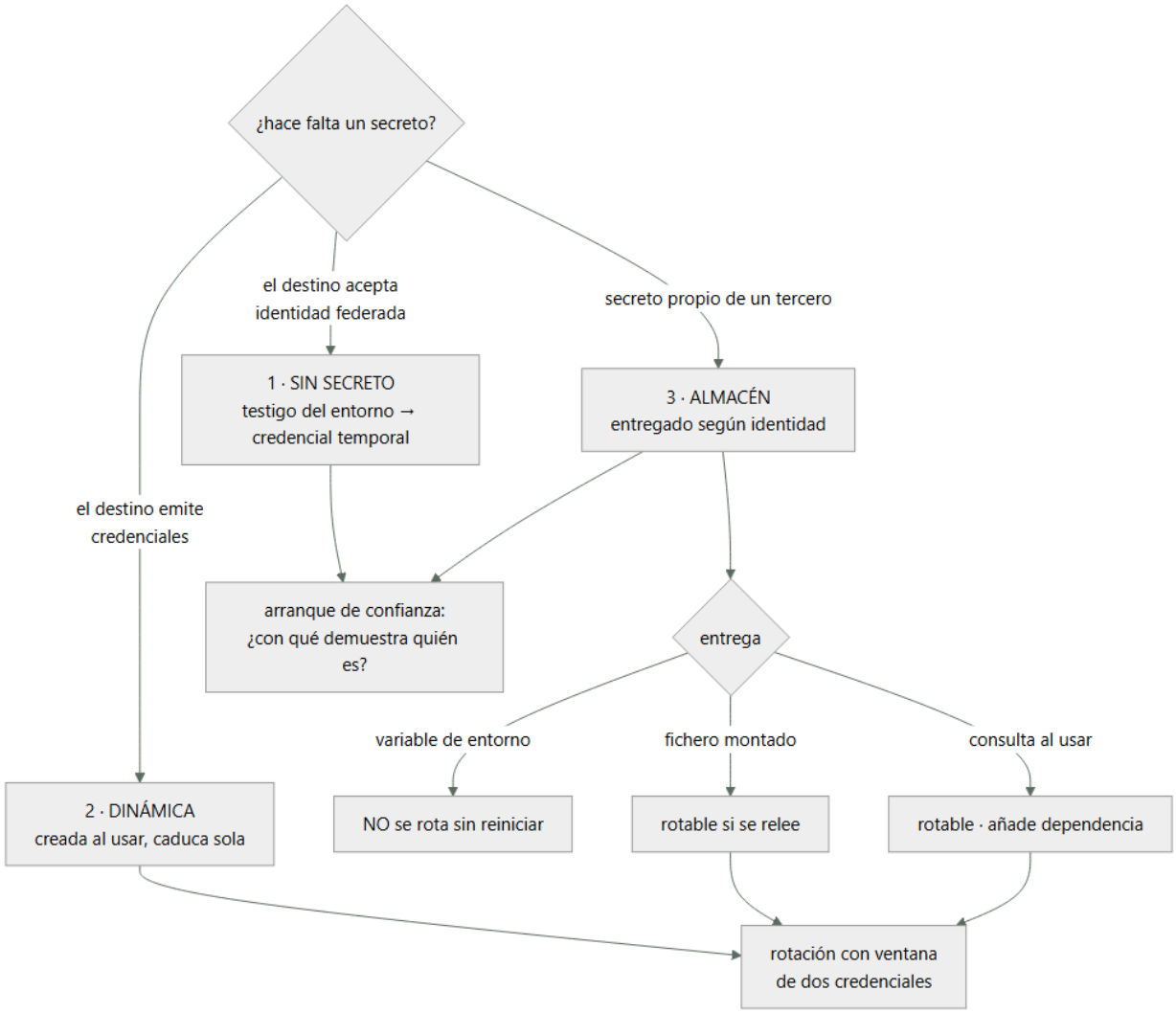
Concepto	Comprensión verificable
identidad federada de carga	La carga demuestra quién es con un testigo emitido por su entorno y obtiene credenciales temporales. No hay secreto que guardar.
credencial dinámica	Credencial creada bajo demanda para un uso concreto, con caducidad corta. Al expirar, deja de existir.
almacén de secretos	Servicio que guarda, versiona, audita y entrega secretos según la identidad que los pide.
entrega en caliente	Que la carga pueda recibir un valor nuevo sin reiniciarse. Es lo que hace posible rotar de verdad.
ventana de dos credenciales	Periodo en el que la antigua y la nueva son válidas a la vez. Es lo que permite rotar sin cortes.
arranque de confianza	Cómo demuestra la carga quién es la primera vez. Si requiere un secreto previo, el problema solo se ha desplazado.

Modelo mental

Gobernar no significa aprobar cada cambio: significa codificar límites, evidencia y responsabilidades para que los equipos puedan avanzar con seguridad.

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. Ordenar por lo que elimina

La escala, de la que hace desaparecer el problema a la que solo lo administra:

1. NO HAY SECRETO
   la carga demuestra quién es con un testigo emitido por su entorno
   y recibe credenciales temporales del destino
   → no hay nada que guardar, rotar ni filtrar
   → clases 054 y 098

2. CREDENCIAL DINÁMICA
   el destino crea una credencial para este uso, con caducidad corta
   → una credencial de base de datos válida 15 minutos, distinta
     en cada arranque
   → si se filtra, caduca sola

3. ALMACÉN DE SECRETOS
   el valor existe y se guarda en un servicio que lo entrega
   según la identidad que lo pide, con auditoría y versiones
   → necesario para terceros que solo tienen claves estáticas

4. CIFRADO EN EL REPOSITORIO
   funciona, y el valor está en un sitio del que no se puede purgar
   → ley 11

5. EN UNA VARIABLE O UN FICHERO DE CONFIGURACIÓN
   → es donde estamos casi siempre al empezar

Y la observación que ordena el trabajo: casi todo el mundo salta al tercero, porque es el que se puede comprar, cuando los dos primeros eliminan categorías enteras de riesgo.

```
con almacén        el secreto existe, se puede filtrar, hay que rotarlo,
                   y quien comprometa la carga lo lee
con federación     no hay secreto; comprometer la carga da acceso
                   MIENTRAS dure, y nada que llevarse
```

Y el criterio para decidir por dónde va cada caso:

```
¿el destino acepta identidad federada?        → opción 1
¿el destino puede emitir credenciales?        → opción 2
¿es un tercero con clave estática y punto?    → opción 3
```

Y la tercera es donde acaba lo que no se puede cambiar: proveedores de pago, mensajería, servicios antiguos. Es una minoría de los casos y suele ocupar toda la atención.

El arranque de confianza, que es la pregunta que descubre si el problema se ha resuelto o solo movido:

```
¿con qué demuestra la carga quién es la PRIMERA vez?

  con un testigo que le da su entorno de ejecución    → resuelto
  con un certificado emitido por la plataforma        → resuelto
  con una clave guardada en su imagen o su
    configuración                                     → NO resuelto:
                                                        hay un secreto
                                                        eterno en algún sitio
```

Y conviene decirlo en voz alta cuando ocurre, en vez de presentar como resuelto un problema que se ha desplazado un escalón.

2. Cómo llega el valor, y por qué eso decide la rotación

Los mecanismos de entrega, con la propiedad que decide todo:

```
VARIABLE DE ENTORNO
  + trivial, funciona en todas partes
  − visible en el listado de procesos, en volcados de memoria,
    y la heredan los procesos hijos
  − NO SE PUEDE CAMBIAR sin reiniciar el proceso

FICHERO MONTADO
  + el contenido puede actualizarse sin reiniciar
  − hay que releerlo: si la aplicación lo lee una vez al arrancar,
    es lo mismo que una variable

CONSULTA AL ALMACÉN EN EL MOMENTO DE USARLO
  + siempre el valor vigente; rotación inmediata
  − añade una dependencia en el camino de la petición
  − se resuelve con caché corta, y entonces la rotación tarda ese caché

AGENTE O ACOMPAÑANTE
  obtiene y renueva por su cuenta y lo deja disponible
  + la aplicación no habla con el almacén
  − una pieza más que operar
```

Y la línea que explica el estado del mundo:

```
si el secreto llega por variable de entorno, rotar exige reiniciar
→ rotar pasa a ser una operación con riesgo
→ y por eso hay credenciales de hace cuatro años
```

Es la ley 16 otra vez: un control que exige una parada acaba no ejecutándose.

Y dos precauciones sobre lo que se lleva el valor a sitios inesperados:

```
los procesos hijos heredan el entorno
  → un guion auxiliar, una herramienta de diagnóstico, un depurador
los volcados y los informes de error suelen incluir el entorno
  → y acaban en el sistema de errores, que ve mucha gente
```

Y la comprobación barata: pedir el entorno de un proceso en producción y ver qué aparece.

```
$ tr '\0' '\n' < /proc/1/environ | grep -Ei 'key|secret|token|password'
```

Si ahí hay algo, ese algo está también en cualquier volcado que se genere.

3. Rotar sin cortar

Rotar significa que el valor antiguo deja de ser válido. Y si el antiguo deja de valer antes de que todo el mundo use el nuevo, hay un corte.

El patrón que lo resuelve son dos credenciales válidas a la vez:

```
1. crear la credencial NUEVA; ahora valen las dos
2. actualizar el almacén con la nueva
3. esperar a que todas las cargas la hayan tomado
   → y comprobarlo, no suponerlo
4. deshabilitar la antigua
5. comprobar que nada falla
6. borrarla
```

Y el paso 3 exige poder observar quién sigue usando la antigua:

```
si el destino registra qué credencial se usó → mirar ahí
si no → esperar más que el mayor caché o ciclo de vida conocido
```

Y los dos errores clásicos:

```
rotar y confiar en que todo se reiniciará    → falla lo que no reinició
deshabilitar la antigua a la vez que se
  publica la nueva                            → corte garantizado
```

Y la rotación automática solo funciona si la carga relee:

```
rotación automática cada 30 días
+ la aplicación lee el valor al arrancar y nunca más
= una caída programada cada 30 días
```

Y hay que decirlo con claridad porque ocurre a menudo: activar la rotación automática sin comprobar la relectura es programar un incidente.

Las credenciales dinámicas eliminan casi todo lo anterior donde se pueden usar:

```
la carga pide acceso a la base
el almacén crea un usuario con permisos concretos y 1 hora de vida
la carga lo usa
al caducar, el usuario se elimina
```

Y lo que se gana:

```
nada que rotar: todo caduca solo
un secreto filtrado sirve una hora
se sabe qué carga hizo qué, porque cada una tiene su usuario
```

Y lo que hay que vigilar: las credenciales huérfanas si la limpieza falla, que es la ley 13 aplicada aquí, y el efecto sobre el agrupador de conexiones de la clase 109, porque cambiar de usuario obliga a rehacer conexiones.

4. El rastro, y qué hacer cuando se filtra

La clase 092 enumeró seis sitios por los que un secreto se escapa. Con lo aprendido desde entonces, la lista es más larga:

```
repositorio e historial                        clase 101
registros de la canalización                   clase 098
artefactos publicados, incluidos ficheros de plan
estado de infraestructura                      clase 087
salidas de módulos
variables de entorno y volcados
registros de la aplicación                     clase 122
informes de error y trazas de pila
telemetría, si se serializan objetos
herramientas de soporte que muestran contexto
capturas de pantalla y mensajes en chats
copias de seguridad de todo lo anterior
```

Y los controles que cubren esa lista ya están repartidos por el programa:

```
protección en el envío del repositorio         clase 101
lista de permitidos en el registro             clase 122
depuración en el recolector de telemetría      clase 124
escáner diario sobre muestras                  clase 122
y el que falta: comprobar qué muestran las herramientas internas
```

Y el procedimiento cuando se filtra, que este programa ha repetido y no cambia:

```
1. ROTAR         estuvo expuesto, punto
2. CORREGIR      el mecanismo que lo dejó salir
3. PURGAR        donde se pueda; sabiendo que no será en todos los sitios
4. REVISAR       qué se hizo con esa credencial mientras estuvo expuesta
```

El cuarto se olvida y es el que dice si hubo daño: con auditoría de uso por credencial se puede responder; sin ella, no.

Y la medida honesta del programa de secretos, que es una sola:

```
tiempo desde que se filtra hasta que deja de ser válida
```

Y lo que la reduce, por orden:

```
no tener secretos                       tiempo = 0
credenciales de una hora                tiempo ≤ 1 h
rotación automática que funciona        horas
rotación manual con procedimiento       días
rotación manual sin procedimiento       nunca
```

Y la lista de comprobación de la clase:

- ■ existe inventario de todos los secretos y de quién los usa
- ■ cada uno está clasificado en la escala, y hay plan para subirlo
- ■ lo que acepta identidad federada no tiene secreto
- ■ las bases de datos usan credenciales dinámicas donde se puede
- ■ el arranque de confianza no depende de un secreto eterno
- ■ ningún secreto llega por variable de entorno si hay que rotarlo
- ■ la aplicación relee el valor sin reiniciar, y está comprobado
- ■ la rotación usa ventana de dos credenciales, y se verifica el paso 3
- ■ hay auditoría de uso por credencial
- ■ el rastro está cubierto en repositorio, registros, telemetría y soporte
- ■ se mide el tiempo desde filtración hasta invalidación
- ■ el procedimiento de filtración empieza por rotar y termina por revisar uso

Y el cierre que enlaza con la clase siguiente: las credenciales son una de las dos vías por las que entra un atacante. La otra es el código y sus dependencias, y el trabajo de mantenerlo sin vulnerabilidades conocidas —con la escala real que eso tiene— es la materia de la clase 138.

Ejemplo trabajado

CloudShop tiene un almacén de secretos desde hace dos años y sigue teniendo credenciales de 2023. El ejercicio empieza por el inventario y termina descubriendo por qué el almacén no había resuelto el problema.

El inventario.

```
secretos en el almacén                                  118
secretos fuera del almacén, encontrados                  41
  en variables de configuración de despliegue            22
  en ficheros de configuración de imágenes               11
  en el estado de infraestructura                         5
  en un documento compartido                              3

edad de la credencial más antigua                   4,1 años
secretos rotados alguna vez                          9 de 159
secretos con dueño identificado                     61 de 159
```

Nueve de ciento cincuenta y nueve rotados alguna vez. El almacén existía y no había cambiado nada, y la razón estaba en el mecanismo de entrega:

```
entrega por variable de entorno                     141 de 159
→ rotar exigía reiniciar
→ reiniciar en producción se percibía como riesgo
→ nadie rotaba
```

La clasificación en la escala.

```
destino acepta identidad federada                88     opción 1: sin secreto
destino puede emitir credenciales dinámicas      34     opción 2
tercero con clave estática                       29     opción 3: almacén
sin clasificar / obsoletos                        8     borrar
```

Ochenta y ocho de ciento cincuenta y nueve no necesitaban existir.

Fase 1: eliminar los 88.

```
credenciales estáticas hacia servicios
del proveedor                                   88              0
mecanismo                              clave de larga     testigo del entorno
                                       duración           → credencial de 1 h
tiempo desde filtración hasta invalidación   indefinido       ≤ 1 h
rotaciones necesarias                           88              0
```

Y el arranque de confianza se revisó explícitamente:

```
¿con qué demuestra la carga quién es?
  en el clúster        con el testigo de su cuenta de servicio
  en la canalización   con el testigo del flujo               clase 098
  en las funciones     con la identidad asignada por la plataforma
→ ningún secreto eterno en el arranque
```

Fase 2: credenciales dinámicas para bases de datos.

```
usuarios de base de datos                  6 fijos      1 por carga y arranque
vida de la credencial                   indefinida            1 h
secretos que rotar                            6               0
quién hizo qué en la base               no se sabía   cada carga, por usuario
```

Y el efecto colateral que hubo que resolver, y que la clase 109 anticipaba:

```
al renovar la credencial, el agrupador tenía que rehacer conexiones
primera versión: rehacía las 24 de golpe cada hora
  → picos de latencia de 400 ms cada 60 minutos
corrección: renovación escalonada, y solapamiento de 10 min
  → pico eliminado
```

Y las credenciales huérfanas, que son la ley 13 aquí:

```
usuarios creados y no eliminados, tras 3 meses                  412
causa            fallos de limpieza cuando la carga moría de golpe
corrección       caducidad en la propia base, no solo en el almacén
                 + barrido diario
usuarios huérfanos después                                      0-3
```

Fase 3: los 29 que no se pueden eliminar.

Aquí sí hacía falta el almacén, y hubo que cambiar la entrega:

```
entrega                              variable de entorno   fichero montado,
                                                           releído al cambiar
rotación                                    manual         automática, 30 días
reinicio necesario                             sí              no
```

Y la primera rotación automática provocó una caída, exactamente como advierte el apartado tercero:

```
02:00  rotación automática del secreto del proveedor de mensajería
02:00  el fichero se actualizó
02:00  la aplicación no lo releía: lo cargaba al arrancar
02:01  fallos en el 100 % de los envíos
02:40  detectado y mitigado reiniciando
```

```
la aplicación relee el valor                    no              sí
comprobado con una prueba                       no        sí, en la canalización
ventana de dos credenciales                     no          sí, 24 h
verificación del paso 3                         no        registro de uso
                                                          por versión
caídas por rotación                      1 de 1 rotación     0 de 14
```

Y la verificación del paso 3 fue lo que dio confianza: el destino registraba qué versión de la credencial se usaba, así que deshabilitar la antigua dejó de ser un acto de fe.

El rastro, revisado con el escáner de la clase 122.

```
hallazgos en la primera semana
  secretos en variables de entorno visibles en volcados            18
  un secreto en el estado de infraestructura                        5
  claves en informes de error con el entorno adjunto                3
  una clave visible en la herramienta interna de soporte            1
  secretos en capturas pegadas en el chat del equipo                2
```

El de la herramienta de soporte era el peor: mostraba la configuración completa de un cliente, incluidas sus claves de integración, y lo veían once personas de atención al cliente.

herramienta de soporte	muestra configuración completa	campos sensibles ocultos
escáner sobre volcados e informes	no	sí, diario
hallazgos residuales	—	0-1 / mes

La medida del programa.

tiempo desde filtración hasta invalidación		
para el 88 %	indefinido	0 (no existen)
para el 21 %	indefinido	≤ 1 h
para el 18 % restante	días o nunca	24 h
secretos totales	159	29
con dueño identificado	61 de 159	29 de 29
rotados en el último año	9 de 159	29 de 29
edad del más antiguo	4,1 años	30 días

A los seis meses.

secretos existentes	159	29
fuera del almacén	41	0
entregados por variable de entorno	141	0
rotados alguna vez	9	29
credenciales dinámicas de base de datos	0	34
usuarios huérfanos en base de datos	—	0-3
caídas por rotación	1	0
secretos visibles en herramientas internas	1	0
tiempo desde filtración hasta invalidación	indefinido	≤ 1 h (o 0)

La lección que esta clase traslada a la parte 11: el almacén de secretos llevaba dos años instalado y el número de credenciales sin rotar era el mismo que antes de comprarlo. La causa no era la herramienta: era que entregarlas por variable de entorno hacía que rotar exigiera reiniciar, y eso convirtió una tarea rutinaria en una operación con riesgo que nadie hacía. Y la mayor reducción del riesgo no la produjo ninguna función del almacén: la produjo descubrir que el 55 % de los secretos no tenía por qué existir.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-11-security-governance-finops/137-gestion-de-secretos-y-credenciales-de-workloads/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 ciclo-secretos 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 ciclo-secretos 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 un almacén de secretos y las credenciales siguen sin rotarse	Se entregan por variable de entorno, así que rotar exige reiniciar	Entrega por fichero releído o por consulta al usar, y comprueba con una prueba que la aplicación toma el valor nuevo sin reiniciar.
La rotación automática provoca una caída	La aplicación lee el valor al arrancar y nunca más	Verifica la relectura antes de activar la rotación y usa ventana con las dos credenciales válidas.
Se gestionan con mucho esfuerzo secretos que no deberían existir	Se saltó a comprar un almacén sin clasificar antes qué destinos aceptan identidad federada o credenciales dinámicas	Clasifica el inventario en la escala y elimina primero todo lo que pueda subir a las dos primeras opciones.
Un secreto aparece en volcados, informes de error o herramientas de soporte	Está en el entorno del proceso y las herramientas muestran configuración completa	Sácalo del entorno, oculta campos sensibles en las herramientas internas y ejecuta un escáner diario sobre muestras.
La base de datos acumula usuarios que ya no usa nadie	Las credenciales dinámicas no se limpian cuando la carga muere de golpe	Pon caducidad en la propia base además del almacén y añade un barrido diario.
Se dice que no hay secretos y hay uno en el arranque	La carga necesita una clave previa para autenticarse ante el almacén	Usa el testigo que emite el entorno de ejecución; si no es posible, dilo explícitamente en vez de presentarlo como resuelto.

■ 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 ordena la escala de opciones y por qué empezar por comprar un almacén no es lo mejor?
2. ¿Por qué una variable de entorno impide rotar y qué consecuencia tiene?
3. ¿Qué pasos tiene una rotación con ventana de dos credenciales y cuál se verifica peor?
4. ¿Qué elimina una credencial dinámica y qué problema nuevo introduce?
5. ¿Cuál es la medida honesta de un programa de secretos?

Referencias

- HashiCorp Vault (2025). Dynamic secrets and database secrets engine — credenciales creadas al usar, con caducidad. <https://developer.hashicorp.com/vault/docs/secrets/databases>
- AWS (2025). Secrets Manager rotation strategies — ventana de dos credenciales y verificación. <https://docs.aws.amazon.com/secretsmanager/latest/userguide/rotating-secrets.html>
- Kubernetes (2025). Secrets: risks and mounted volumes — entrega por fichero frente a variable de entorno. <https://kubernetes.io/docs/concepts/configuration/secret/>
- SPIFFE (2025). Workload identity and the bootstrap problem — cómo demuestra una carga quién es sin un secreto previo. <https://spiffe.io/docs/latest/spiffe-about/overview/>

- OWASP (2025). Secrets management cheat sheet — inventario, entrega, rotación y respuesta a filtraciones. <https://cheatsheetseries.owasp.org/cheatsheets/SecretsManagementCheatSheet.html>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 11 en PDF · Manual integral

Anterior	Índice	Siguiente
← 136 · Cifrado, KMS, HSM, rotación y envelope encryption	Parte 11 · Programa	138 · Vulnerabilidades, imágenes y cadena de suministro →

Evaluación — 137 Gestión de secretos y credenciales de workloads

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 ciclo-secretos 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.

138 — Vulnerabilidades, imágenes y cadena de suministro

Parte: 11 — Seguridad, gobierno, cumplimiento y FinOps Nivel: avanzado · Horas estimadas: 4 Laboratorio: supply-chain · Estado: EXECUTABLECORE

Propósito

Gobernar las vulnerabilidades de lo que se ejecuta, con la escala real del problema delante: una organización mediana tiene decenas de miles de hallazgos y capacidad para corregir unos cientos al trimestre. Por eso la clase no trata de escanear —eso ya lo montó la 101— sino de priorizar con criterios que reduzcan de verdad, de corregir por familias en vez de una a una, y de la parte que ni la 067 ni la 101 cubrieron: los ataques que llegan por la propia dependencia, empezando por el más barato de ejecutar y el más fácil de prevenir.

Resultados de aprendizaje

Al finalizar podrás:

1. Priorizar por explotabilidad y exposición, no por gravedad nominal.
2. Corregir por familias: base, directas, transitivas y código propio.
3. Prevenir la sustitución de dependencias por paquetes ajenos.
4. Fijar plazos y excepciones que se puedan cumplir.
5. Medir el programa por antigüedad de lo pendiente, no por número de hallazgos.

Conceptos centrales

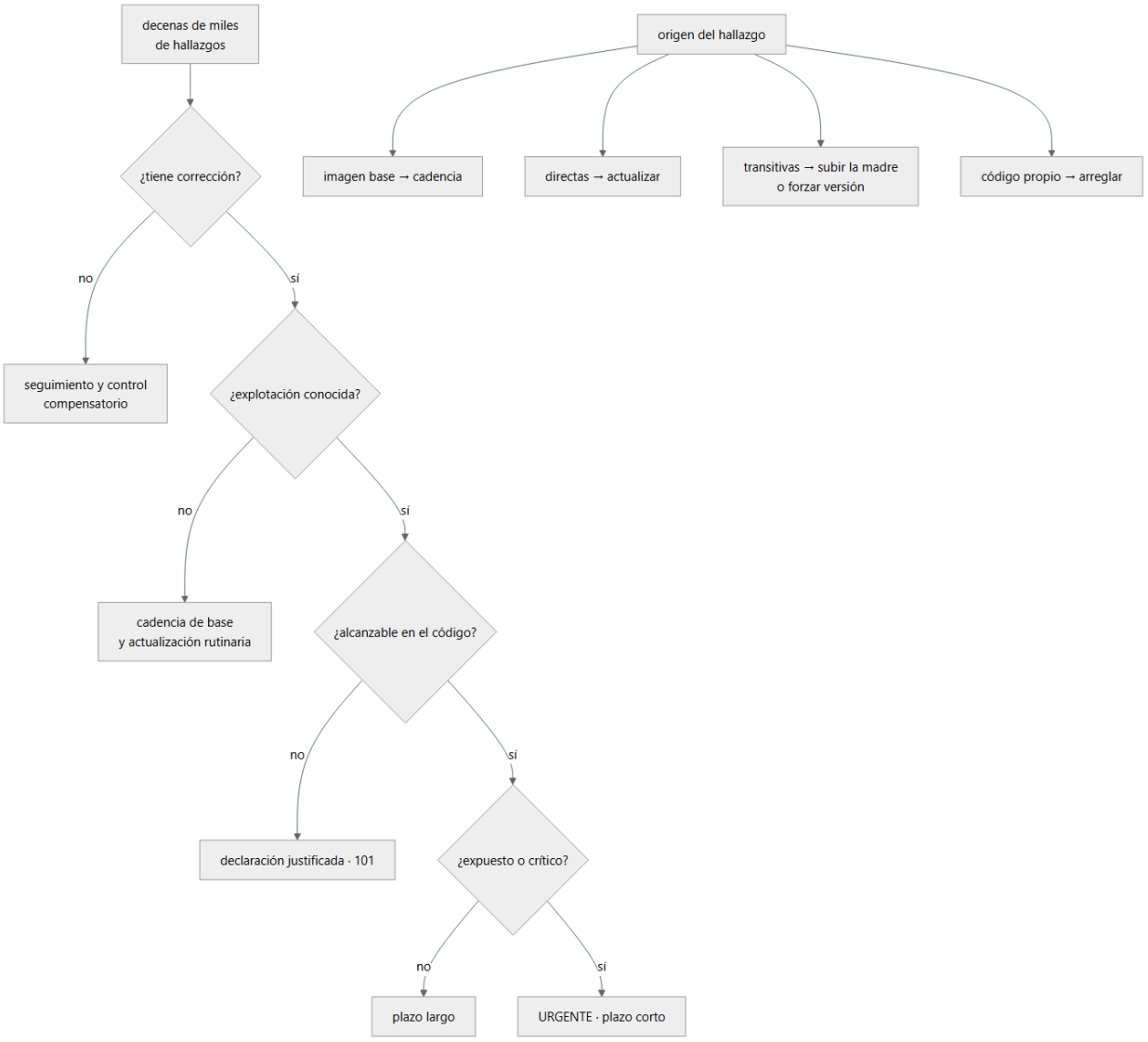
Concepto	Comprensión verificable
explotabilidad conocida	Que exista explotación real observada. Es el filtro que más reduce y el que mejor se correlaciona con el riesgo.
exposición	Si el componente vulnerable es alcanzable desde fuera o solo desde dentro. Cambia la prioridad más que la gravedad nominal.
dependencia transitiva	La que llega a través de otra. Es la mayoría del recuento y no se puede actualizar directamente.
confusión de dependencias	Ataque en el que un paquete público con el nombre de uno interno se instala en lugar del legítimo.
cadencia de base	Reconstruir con imagen base actualizada de forma periódica. Elimina en bloque la mayoría de los hallazgos sin analizarlos.
antigüedad de lo pendiente	Tiempo que lleva sin corregir el hallazgo más viejo que cumple los criterios de prioridad. Es la medida útil del programa.

Modelo mental

Gobernar no significa aprobar cada cambio: significa codificar límites, evidencia y responsabilidades para que los equipos puedan avanzar con seguridad.

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 escala obliga a priorizar

Los números de una organización mediana, para no engañarse:

```
hallazgos totales en imágenes y dependencias        30.000 - 80.000
capacidad real de corrección al trimestre               200 - 600
```

Con esa proporción, la pregunta no es cómo corregir más, sino cómo elegir qué. Y el criterio habitual —la gravedad nominal— es malo:

```
hallazgos de gravedad crítica                          ~3.000
de ellos, con explotación real observada                  ~40
```

La gravedad describe el impacto si se explota, no la probabilidad de que ocurra. Los cuatro filtros que sí reducen, en orden de eficacia:

1. ¿HAY CORRECCIÓN?
   sin versión corregida, no es una tarea: es un seguimiento

2. ¿HAY EXPLOTACIÓN CONOCIDA?
   existen catálogos públicos de vulnerabilidades explotadas de verdad
   → es el filtro que más reduce, con diferencia

3. ¿ES ALCANZABLE?
   el código vulnerable, ¿se ejecuta desde esta aplicación?   clase 101

4. ¿ESTÁ EXPUESTO O ES CRÍTICO?

```
   accesible desde fuera, o toca datos personales, o es el camino de pago
```

Y el embudo, con números realistas:

```
todos                       47.000
con corrección              21.000
con explotación conocida       310
alcanzables                    140
expuestos o críticos            61
```

Sesenta y uno es un número con el que se puede trabajar; cuarenta y siete mil no.

Y los dos errores simétricos que evita este embudo:

```
tratar los 47.000 como pendientes    → parálisis, y el escáner acaba
                                       desactivado (ley 16)
ignorarlos todos porque son muchos   → los 61 tampoco se arreglan
```

Y la medida del programa, que no es el recuento:

```
mal    «tenemos 47.000 hallazgos» → no dice nada y no baja nunca
bien   antigüedad del más viejo que supera el embudo
       → si el más antiguo tiene 9 días, el programa funciona
       → si tiene 7 meses, no
```

Y conviene además vigilar la tendencia del embudo entero, porque una entrada que crece más deprisa que la salida es un problema que se ve un año antes de doler.

2. Corregir por familias

Los hallazgos vienen de sitios distintos y cada uno se corrige de una forma distinta. Tratarlos igual es lo que hace inabordable el trabajo.

```
IMAGEN BASE                          suele ser el 70-90 % del recuento
  → no se analizan uno a uno: se RECONSTRUYE con la base al día
  → una cadencia mensual elimina la mayoría sin tomar ninguna decisión

DEPENDENCIAS DIRECTAS                las que tú declaraste
  → actualizar; es trabajo normal y se automatiza con propuestas de cambio

DEPENDENCIAS TRANSITIVAS             la mayoría del recuento de código
  → no se pueden actualizar directamente

EL LENGUAJE O TIEMPO DE EJECUCIÓN
  → actualizaciones mayores, planificadas

CÓDIGO PROPIO                        clase 101
  → puerta en modo diferencial
```

Y la primera línea es la que más rinde: reconstruir periódicamente es la palanca más grande de todo el programa, y no requiere priorizar nada.

```
sin cadencia   los hallazgos se acumulan hasta que alguien reacciona
con cadencia   la base entra al día en cada reconstrucción

efecto medido habitual: entre el 60 % y el 85 % del recuento desaparece
```

Y lo que la hace posible: imágenes mínimas. Cuantos menos paquetes contiene una imagen, menos hallazgos tendrá, y esa reducción es permanente:

```
imagen de distribución completa      ~450 paquetes
imagen mínima                         ~90
imagen sin distribución                ~15
```

Las transitivas, que es el caso incómodo. Tres opciones y ninguna es cómoda:

```
SUBIR LA DEPENDENCIA MADRE
  la correcta; y a veces la madre no ha publicado versión

FORZAR LA VERSIÓN de la transitiva
  funciona en casi todos los gestores modernos
  → y puedes estar poniendo una versión que la madre no ha probado
  → hay que ejecutar las pruebas y anotar por qué se forzó

SUSTITUIR LA DEPENDENCIA MADRE
  caro, y a veces es la respuesta correcta si no se mantiene
```

Y el fichero de versiones fijadas es la fuente de verdad: lo que se instala es lo que dice ese fichero, no lo que dice el manifiesto. Si no está fijado con versiones exactas, dos construcciones del mismo código instalan cosas distintas, y eso rompe la inmutabilidad de la clase 099.

Y una decisión de higiene que reduce el trabajo futuro:

```
cada dependencia nueva se justifica en la revisión
y se revisa: ¿está mantenida? ¿cuántas dependencias arrastra?
→ una dependencia con 140 transitivas es una decisión, no un detalle
```

3. Cuando el ataque llega por la dependencia

Las clases 067 y 101 se ocuparon de saber qué contiene un artefacto y de verificar su origen. Falta lo que ocurre antes: que lo que se instala no sea lo que se cree.

```
CONFUSIÓN DE DEPENDENCIAS
  tu organización tiene un paquete interno llamado «tienda-utils»
  alguien publica «tienda-utils» en el repositorio PÚBLICO
  con un número de versión más alto
  → muchos gestores prefieren la versión más alta
  → y en la siguiente construcción se instala el del atacante
```

Es el ataque más barato de ejecutar de esta lista y el más fácil de prevenir:

```
reservar los nombres internos en los repositorios públicos
configurar el gestor para que los paquetes del ámbito interno
  SOLO se busquen en el repositorio interno
usar un ámbito o prefijo propio para todo lo interno
y prohibir la resolución mixta entre repositorios
```

Y los demás:

```
NOMBRE PARECIDO           un paquete llamado casi igual que uno popular
  → se previene con lista de permitidos y revisión de dependencias nuevas

CUENTA DE MANTENEDOR COMPROMETIDA
  la versión nueva de un paquete legítimo trae código malicioso
  → se mitiga fijando versiones con huella, y no actualizando
    automáticamente sin ningún retraso

GUIONES DE INSTALACIÓN
  se ejecuta código del paquete durante la instalación,
  con los permisos de quien construye
  → desactivarlos donde el gestor lo permita

COMPROMISO DEL PROPIO SISTEMA DE CONSTRUCCIÓN
  → es la clase 098: identidad del flujo, permisos mínimos y procedencia
```

Y el control que cubre a la vez varios de estos, y que además resuelve problemas de disponibilidad:

```
REPOSITORIO INTERNO COMO ÚNICO ORIGEN
  las construcciones no descargan de internet: descargan del interno
  el interno replica lo aprobado desde el público
  → permite lista de permitidos, retención y análisis previo
  → y una construcción no falla porque un repositorio público esté caído
```

Y dos prácticas que suenan contradictorias y no lo son:

```
FIJAR con huella, para que lo que se instala sea siempre lo mismo
ACTUALIZAR con cadencia, para no acumular vulnerabilidades
→ y entre las dos, un RETRASO deliberado de unos días en actualizar
  versiones recién publicadas, que es cuando se detectan los paquetes
  comprometidos
```

4. Plazos, excepciones y lo que no tiene arreglo

Los plazos se fijan por el embudo, no por la gravedad nominal:

```
explotable + alcanzable + expuesto        7 días
explotable + alcanzable                  30 días
explotable, no alcanzable                90 días o declaración justificada
con corrección, sin explotación conocida  cadencia normal de actualización
sin corrección                            seguimiento, sin plazo
```

Y lo que hace que los plazos signifiquen algo, con la disciplina de las clases 046, 091 y 101:

```
cada incumplimiento produce una excepción, no un silencio
```

```
con motivo, responsable y caducidad
y la caducidad rompe la canalización
```

Y una decisión organizativa que decide si el programa funciona: quién puede aceptar el riesgo. Un hallazgo que supera el embudo y no se corrige en plazo escala a alguien que puede decidir que se acepta, con su nombre.

Cuando no hay corrección disponible, que ocurre a menudo:

```
controles compensatorios
  quitar la exposición: sacar el componente de internet
  bloquear la ruta vulnerable en el filtro de aplicación   clase 135
  restringir permisos de esa carga                         clase 134
  desactivar la funcionalidad afectada                     clase 105
y seguimiento con revisión periódica, hasta que haya versión
```

La primera es la más eficaz y la que menos se considera: un componente vulnerable que no es alcanzable desde fuera cambia de categoría entera.

Y lo que hay que vigilar de forma continua:

```
antigüedad del hallazgo más viejo que supera el embudo
entradas frente a salidas del embudo, por semana
edad media de las imágenes base en producción
proporción de artefactos reconstruidos en los últimos 30 días
dependencias nuevas añadidas, y por quién
excepciones vivas y su caducidad
y lo que la clase 101 exigía: reescaneo de lo DESPLEGADO, no de lo construido
```

Y la lista de comprobación de la clase:

- existe un embudo escrito y se aplica antes de asignar trabajo
- se usa un catálogo de explotación real, no solo la gravedad
- hay cadencia de reconstrucción con base actualizada
- las imágenes son mínimas
- las versiones están fijadas con huella
- hay retraso deliberado antes de adoptar versiones recién publicadas
- los nombres internos están reservados en los repositorios públicos
- los ámbitos internos solo se resuelven en el repositorio interno
- las construcciones descargan del repositorio interno, no de internet
- los guiones de instalación están desactivados donde se pueda
- los plazos salen del embudo y las excepciones caducan
- la medida publicada es la antigüedad de lo pendiente, no el recuento
- se reescanea lo desplegado, no solo lo que se construye

Y el cierre que enlaza con la clase siguiente: hasta aquí, controles sobre lo que se construye y se despliega. Queda saber si la configuración de lo que ya está en marcha cumple lo que se supone, detectarlo continuamente y corregirlo sin abrir mil tareas, que es la materia de la clase 139.

Ejemplo trabajado

CloudShop tiene 47.000 hallazgos abiertos y un equipo que puede corregir unos 400 al trimestre. El ejercicio consiste en construir el embudo, corregir por familias y prevenir el ataque que estuvo a punto de ocurrir.

El embudo, aplicado.

```
hallazgos totales                                  47.180
con versión corregida disponible                   21.340
con explotación real observada                        310
alcanzables desde el código                           142
expuestos a internet o en el camino de pago            61
```

Sesenta y uno frente a cuarenta y siete mil. Y el reparto de los sesenta y uno:

```
en dependencias transitivas                            38
en dependencias directas                               14
en el tiempo de ejecución                               6
en la imagen base                                       3
```

Corrección por familias, y lo que aportó cada una.

```
CADENCIA DE BASE (mensual, automatizada)
  hallazgos antes de la primera reconstrucción         47.180
  después                                              11.900
  reducción                                              75 %
  esfuerzo                                  un flujo programado
```

Setenta y cinco por ciento del recuento eliminado sin analizar ni un solo hallazgo.

```
IMAGEN MÍNIMA
  paquetes en la imagen anterior                          438
  paquetes en la imagen mínima                             94
  hallazgos tras el cambio                              4.200
  reducción adicional                                     65 %
  coste     3 servicios necesitaron herramientas que no estaban;
            se añadieron explícitamente

DIRECTAS
  propuestas de actualización automáticas, agrupadas por semana
  actualizadas en 3 meses                                 184
  roturas causadas                                          7
  todas detectadas por las puertas de la clase 100

TRANSITIVAS, las 38 del embudo
  resueltas subiendo la dependencia madre                  23
  resueltas forzando la versión                            11
    → con las pruebas ejecutadas y el motivo anotado
  sin versión corregida disponible                          4
    → control compensatorio: 3 se sacaron de internet,
      1 se bloqueó en el filtro de aplicación
```

El estado del embudo a los tres meses.

```
hallazgos totales                       47.180          3.100
con corrección                          21.340          1.410
con explotación conocida                   310             22
alcanzables                                142             11
expuestos o críticos                        61              2
antigüedad del más viejo del embudo    no se medía        6 días
```

Y la tendencia, que es la que dice si esto se sostiene:

```
entradas al embudo por semana                             4-9
salidas por semana                                       8-14
→ la salida supera a la entrada
```

La confusión de dependencias, encontrada por casualidad.

Al montar el repositorio interno, alguien revisó qué nombres internos existían también en los repositorios públicos:

```
paquetes internos                                          34
con el mismo nombre publicado en un repositorio público     3
  de ellos, publicados por la propia empresa años atrás     1
  de ellos, publicados por terceros desconocidos            2
```

Y uno de los dos era claramente un intento:

```
nombre                    igual que el paquete interno de utilidades
versión publicada         99.9.9   ← más alta que cualquier interna
publicado hace            4 meses
descargas registradas     1.180
contenido                 guion de instalación que enviaba variables
                          de entorno a un servidor externo
¿llegó a instalarse aquí? no, por suerte: el repositorio interno estaba
                          configurado como origen preferente
                          en 11 de 15 servicios
```

Cuatro servicios estaban a un despliegue de instalarlo.

```
nombres internos reservados en repositorios
públicos                                    0 de 34        34 de 34
ámbito propio para todo lo interno            no             sí
resolución mixta entre repositorios           sí             no
construcciones que descargan de internet    15 de 15       0 de 15
guiones de instalación                     permitidos    desactivados
                                                         (2 excepciones
                                                          con motivo)
```

Y los 1.180 descargas registradas por el paquete falso indican que otras organizaciones sí lo instalaron.

El retraso deliberado, y la vez que sirvió.

```
política adoptada    no adoptar versiones con menos de 5 días publicadas,
                     salvo correcciones de seguridad urgentes
```

```
mes 5   un paquete popular publica una versión con código malicioso
        introducido por una cuenta de mantenedor comprometida
        retirado por el repositorio a las 31 horas
        CloudShop no la había adoptado: faltaban 3 días de la ventana
```

Los plazos y las excepciones.

```
plazos definidos                              no        4 niveles
excepciones vivas                       no se sabía         9
con motivo, responsable y caducidad           —          9 de 9
caducadas y renovadas                         —             2
aceptaciones de riesgo firmadas               0             3
```

Las tres aceptaciones firmadas son componentes sin versión corregida en un sistema heredado, con controles compensatorios y revisión trimestral.

El reescaneo de lo desplegado, de la clase 101.

```
primera ejecución tras el nuevo embudo
  huellas en producción                                  18
  hallazgos que superaban el embudo                       2
  ambos publicados después de la última construcción
  → ninguna canalización los habría detectado
```

A los seis meses.

```
hallazgos totales                          47.180          2.400
que superan el embudo                          61              1
antigüedad del más viejo del embudo     no se medía        4 días
paquetes por imagen                            438            94
cadencia de reconstrucción                  ninguna        mensual
artefactos reconstruidos en 30 días           12 %           100 %
nombres internos reservados                 0 de 34        34 de 34
construcciones que descargan de internet    15 de 15         0 de 15
versiones fijadas con huella                 3 de 15        15 de 15
intentos de confusión de dependencias
detectados                                     —              2
excepciones con caducidad                      0              9
```

La lección que esta clase traslada a la parte 11: el 75 % de los cuarenta y siete mil hallazgos desapareció con un flujo programado que reconstruye las imágenes cada mes, sin que nadie priorizara nada; y otro 65 % del resto, con una imagen que contiene ochenta y cuatro paquetes menos. Priorizar hacía falta para sesenta y uno, no para cuarenta y siete mil. Y el riesgo más serio del semestre no fue ninguna vulnerabilidad de la lista: fue un paquete con el nombre de uno interno, versión 99.9.9, esperando en un repositorio público a que cuatro servicios se construyeran.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-11-security-governance-finops/138-vulnerabilidades-imagenes-y-cadena-de-
suministro/lab.py
```

El laboratorio selecciona el motor de práctica supply-chain 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 informe-supply-chain en evidence/ usando la plantilla indicada.
5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

El artefacto principal es procedencia, inventario y verificación del artefacto. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye informe-supply-chain 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 decenas de miles de hallazgos y no se corrige ninguno	Se prioriza por gravedad nominal, que no distingue lo explotable de lo que no	Aplica el embudo: con corrección, con explotación conocida, alcanzable, expuesto o crítico.
El recuento no baja aunque se trabaje mucho	Se corrigen hallazgos uno a uno cuando la mayoría vienen de la imagen base	Reconstruye con cadencia y usa imágenes mínimas; eso elimina la mayor parte sin analizar nada.
Dos construcciones del mismo código instalan versiones distintas	No hay fichero de versiones fijadas con huella	Fija versiones exactas con huella y trata ese fichero como la fuente de verdad.
Un paquete público con el nombre de uno interno acaba instalado	Resolución mixta entre repositorios y preferencia por la versión más alta	Reserva los nombres internos, usa un ámbito propio, resuelve lo interno solo en el repositorio interno y construye desde él.
Una versión recién publicada de una dependencia introduce código malicioso	Se actualiza automáticamente sin ninguna ventana	Retraso deliberado de unos días antes de adoptar versiones nuevas, salvo correcciones urgentes de seguridad.
Se publica el número de hallazgos como medida y nadie sabe si mejora	El recuento no distingue lo urgente de lo irrelevante	Publica la antigüedad del hallazgo más viejo que supera el embudo y la relación entre entradas y salidas.

■ 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 cuatro filtros componen el embudo y cuál reduce más?
2. ¿Por qué la cadencia de reconstrucción es la palanca mayor del programa?
3. ¿Qué tres opciones hay ante una vulnerabilidad en una dependencia transitiva?
4. ¿Cómo funciona la confusión de dependencias y cómo se previene?
5. ¿Por qué la medida útil es la antigüedad de lo pendiente y no el recuento?

Referencias

- CISA (2025). Known Exploited Vulnerabilities catalog — priorizar por explotación real observada. <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>
- FIRST (2025). EPSS: exploit prediction scoring system — probabilidad de explotación frente a gravedad. <https://www.first.org/epss/>
- Birsan, A. (2021). Dependency confusion — descripción del ataque y su prevención. <https://medium.com/@alex.birsan/dependency-confusion-4a5d60fec610>
- OpenSSF (2025). Supply chain best practices: pinning, allow-listing and internal mirrors — controles sobre lo que se instala. <https://best.openssf.org/>
- Google (2025). Distroless container images — imágenes mínimas y su efecto en el recuento de hallazgos. <https://github.com/GoogleContainerTools/distroless>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 11 en PDF · Manual integral

Anterior	Índice	Siguiente
← 137 · Gestión de secretos y credenciales de workloads	Parte 11 · Programa	139 · CSPM, postura, policy as code y remediación →

Evaluación — 138 Vulnerabilidades, imágenes y cadena de suministro

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 informe-supply-chain 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.

139 — CSPM, postura, policy as code y remediación

Parte: 11 — Seguridad, gobierno, cumplimiento y FinOps Nivel: avanzado · Horas estimadas: 4 Laboratorio: governance · Estado: EXECUTABLECORE

Propósito

Comprobar que la configuración de lo que ya está en marcha cumple lo que se supone, que no es lo mismo que comprobar lo que se iba a crear. La clase sitúa los tres momentos en que se puede aplicar un control y sostiene que el más fuerte es el que no se puede rodear; afronta el problema que aparece siempre al encender esta clase de herramientas —miles de hallazgos y ningún dueño—; y trata la corrección automática con la desconfianza que merece, porque es un cambio en producción que nadie revisó.

Resultados de aprendizaje

Al finalizar podrás:

1. Elegir el momento de aplicar cada control: proveedor, canalización o detección.
2. Reducir el ruido inicial con línea base y criterios de exposición.
3. Atribuir cada hallazgo a un dueño, sin lo cual no es una tarea.
4. Escribir reglas como código, con prueba negativa.
5. Corregir en el origen y no en el recurso, y acotar la corrección automática.

Conceptos centrales

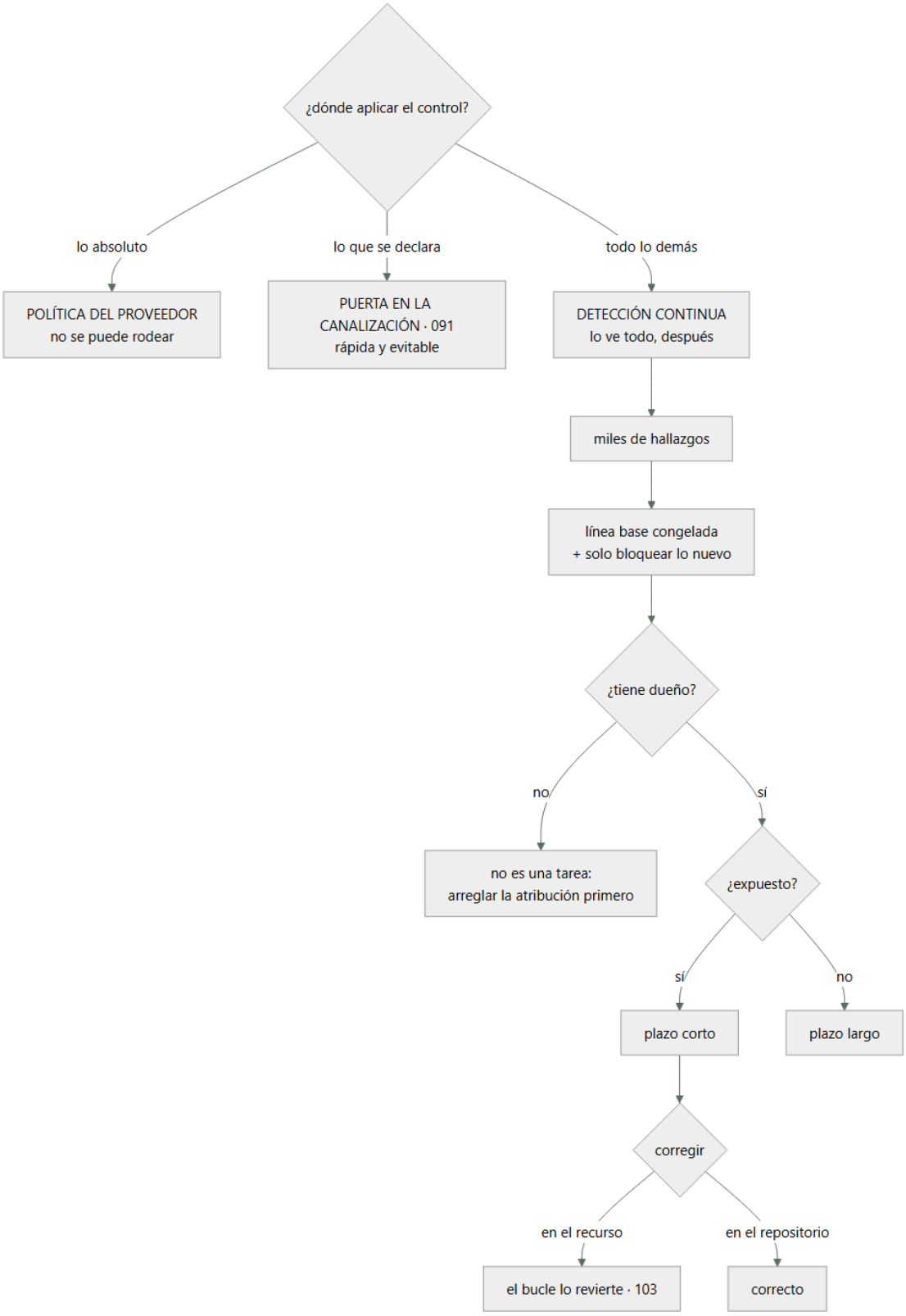
Concepto	Comprensión verificable
postura	Estado real de configuración de todo lo desplegado, medido de forma continua contra un conjunto de reglas.
control preventivo del proveedor	Política de organización que impide la acción, aunque se intente a mano y con permisos. Es el único que no se rodea.
línea base	Conjunto de hallazgos existentes que se congela para poder bloquear solo lo nuevo. Es lo que hace adoptable la herramienta.
atribución	Saber de quién es cada recurso. Sin ella un hallazgo no se puede asignar y se acumula.
corrección en el origen	Arreglar la declaración en el repositorio, no el recurso. Si se toca el recurso, el bucle lo revierte.
corrección automática	Cambio aplicado sin intervención humana. Necesita las mismas garantías que cualquier despliegue, y contador.

Modelo mental

Gobernar no significa aprobar cada cambio: significa codificar límites, evidencia y responsabilidades para que los equipos puedan avanzar con seguridad.

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 momentos, y cuál gana

Un mismo control se puede aplicar en tres sitios, y no son equivalentes:

1. `POLÍTICA DEL PROVEEDOR`
   `la acción no se puede ejecutar, ni a mano, ni con permisos de administrador`
   `+ es el único control que NO se puede rodear`

```
   – rígido: si es demasiado amplio, bloquea trabajo legítimo
   ejemplos  «ninguna cuenta puede desactivar el registro de auditoría»
             «ningún almacén puede hacerse público»
             «ninguna región fuera de las autorizadas»

2. PUERTA EN LA CANALIZACIÓN                              clase 091
   se analiza lo que se va a crear y se bloquea antes de aplicar
   + rápido, con buen mensaje de error, y educa
   – solo ve lo que pasa por ahí

3. DETECCIÓN CONTINUA SOBRE LO DESPLEGADO   ← esta clase
   se observa el estado real
   + lo ve ABSOLUTAMENTE todo: lo creado a mano, lo que cambió el proveedor,
     lo que existe desde antes
   – llega después de que ocurra
```

Y la regla que ordena la elección:

```
lo que NUNCA debe ocurrir           → política del proveedor
lo que casi nunca debe ocurrir      → puerta en la canalización
todo lo demás, y para comprobar
que los dos anteriores funcionan    → detección continua
```

Y la última mitad de esa tercera línea es la más olvidada: la detección es también la comprobación de que los controles preventivos existen y siguen activos.

Y el motivo por el que la detección hace falta aunque todo se despliegue con infraestructura declarada:

```
recursos creados a mano en una urgencia y nunca declarados
recursos creados por servicios que crean otros recursos
  → un clúster que crea discos, un servicio que crea reglas de red
cambios en los valores por defecto del proveedor
recursos heredados de antes de la infraestructura declarada
recursos de cuentas que nadie sabía que existían
```

La última es más frecuente de lo que parece, y explica por qué la primera medida de esta clase es la cobertura:

```
¿cuántas cuentas o proyectos hay?
¿cuántos están siendo analizados?
→ si la respuesta no es «todos», el resto de las cifras no significan nada
```

2. Miles de hallazgos y ningún dueño

Al activar la detección aparecen entre cientos y decenas de miles de hallazgos. Y la historia que sigue ya la conoce este programa: ley 15, y a los ocho meses nadie mira el panel.

La medicina es la misma de la clase 101, con una diferencia importante:

```
1. LÍNEA BASE congelada, con su cifra publicada
2. bloquear o alertar SOLO por lo nuevo
3. objetivo de reducción de la base, con dueño y trimestre
4. priorizar por EXPOSICIÓN, no por gravedad nominal
```

Y el criterio de exposición, que es el que de verdad ordena:

```
accesible desde internet                          máxima prioridad
contiene o toca datos personales
permite obtener credenciales o escalar permisos   ← clase 133
afecta al registro de auditoría o a las copias
todo lo demás
```

Y la diferencia con la clase 101 es la que decide si el programa avanza: aquí el hallazgo está sobre un recurso, y hay que saber de quién es.

```
un hallazgo sin dueño no es una tarea: es una queja
```

Y la atribución se resuelve con lo que este programa ya tiene:

```
etiquetas obligatorias en la creación: servicio, equipo, entorno
  → impuestas por política del proveedor, no por buena voluntad
catálogo de servicios con su equipo dueño          clase 095
y para lo que no tenga etiqueta: mirar quién lo creó en el
  registro de auditoría
```

Y el orden importa: antes de repartir hallazgos, hay que poder repartirlos. Un programa que empieza asignando trabajo sin atribución fiable acaba con todo en la cola de un equipo de plataforma.

Y lo que hay que medir sobre la propia atribución:

```
recursos sin etiqueta de dueño
recursos cuyo equipo no existe en el catálogo
recursos que no aparecen en ninguna declaración de infraestructura
```

La tercera merece su propia campaña: lo que no está declarado no se puede corregir en el origen, que es el apartado cuarto.

3. Reglas como código

Las reglas del catálogo por defecto sirven para empezar y no bastan: las que más valen son las propias de la organización.

```
del catálogo   cifrado activo, almacenes no públicos, registro habilitado,
               versiones de protocolo, puertos administrativos abiertos
propias        «todo recurso tiene etiqueta de equipo y de entorno»
               «ninguna base de producción admite conexiones desde dev»
               «las copias de seguridad tienen retención mínima de 30 días»
               «ningún almacén con datos personales fuera de estas regiones»
```

Y las reglas se tratan como código, con lo que eso implica:

```
viven en un repositorio, con revisión
se despliegan por la canalización
se versionan, y un cambio de regla es un cambio
y cada una tiene PRUEBA NEGATIVA
```

La última es la que separa una regla de un deseo:

```
una prueba que crea a propósito un recurso que la incumple
y comprueba que la regla lo detecta
→ sin eso, no se sabe si la regla funciona o si está mal escrita
→ y una regla que no detecta nada parece que todo va bien: ley 13
```

Y una comprobación periódica sobre el conjunto:

```
reglas que no han producido ningún hallazgo en 6 meses
  → ¿es que se cumple siempre, o es que la regla está rota?
  → se resuelve con la prueba negativa, no suponiendo
```

Y la excepción, con la disciplina de siempre:

```
en el código de la regla o junto al recurso
con motivo, responsable y caducidad
y la caducidad reabre el hallazgo
```

Y un detalle práctico que evita discusiones: la excepción se declara donde se declara el recurso, no en la consola de la herramienta. Así viaja con el código y se revisa con él.

4. Corregir sin romper

Dónde se corrige es la decisión que más problemas evita:

```
mal   modificar el recurso directamente
      → si está gobernado por un bucle de reconciliación, se revierte
      → y queda una pelea entre dos automatismos              clase 103

bien  corregir la declaración en el repositorio
      → un cambio propuesto automático, con el arreglo hecho
      → revisado y aplicado por el camino normal
```

Y para lo que no está declarado, hay que decidir entre dos cosas y ninguna es «tocarlo y ya»:

```
importarlo a la infraestructura declarada y corregirlo ahí
o eliminarlo, si no debería existir
```

La corrección automática es tentadora y hay que tratarla como lo que es: un cambio en producción que nadie revisó.

```
seguro de automatizar
  añadir una etiqueta que falta
  cerrar el acceso público de un almacén recién creado
  desactivar una credencial no usada en 180 días
  borrar reglas de red que no permiten nada

NO automatizar
  cambios que puedan cortar tráfico legítimo
```

```
  modificar permisos de una identidad en uso
  borrar recursos con estado
  nada cuyo efecto no se pueda probar antes
```

Y las garantías que necesita, que son las de la parte 08 y la ley 19:

```
primero en modo simulación: qué habría cambiado
después en entornos inferiores
escalonado, no en todas las cuentas a la vez
contador y límite: si actúa más de N veces, se detiene y avisa
registro de cada actuación, con quién, qué y por qué
y forma de deshacerlo
```

Y el contador es imprescindible por el motivo de la clase 132: una corrección automática que actúa doscientas veces al día está tapando que algo genera esos recursos mal configurados una y otra vez.

Y lo que hay que vigilar en el programa completo:

```
cobertura: cuentas analizadas frente a cuentas existentes
hallazgos nuevos por semana, y su tendencia
antigüedad del más viejo entre los expuestos
recursos sin dueño
recursos fuera de la infraestructura declarada
excepciones vivas y caducadas
actuaciones de corrección automática, por regla
reglas sin hallazgos en 6 meses
```

Y la lista de comprobación de la clase:

- lo que nunca debe ocurrir está en política del proveedor, no en un aviso
- la detección cubre todas las cuentas y proyectos, y se comprueba
- hay línea base congelada con cifra publicada y objetivo de reducción
- la prioridad se decide por exposición, no por gravedad nominal
- las etiquetas de dueño son obligatorias en la creación
- todo hallazgo se asigna a un equipo del catálogo
- las reglas viven como código, con revisión y prueba negativa
- las excepciones se declaran junto al recurso y caducan
- la corrección se hace en el repositorio, no en el recurso
- lo no declarado se importa o se elimina
- la corrección automática se limita a lo que no puede cortar tráfico
- toda corrección automática tiene contador, límite y registro
- se mide la cobertura y la antigüedad de lo expuesto

Y el cierre que enlaza con la clase siguiente: todo lo anterior comprueba reglas conocidas. Lo que no cubre es preguntarse qué intentaría alguien contra este sistema en concreto, que es un trabajo distinto y no automatizable, y es la materia de la clase 140.

Ejemplo trabajado

CloudShop activa detección continua de configuración sobre sus cuentas. Lo primero que descubre no es un problema de seguridad: es que no sabe cuántas cuentas tiene.

La cobertura, antes que nada.

```
cuentas y proyectos en el inventario documentado              14
cuentas encontradas en la facturación consolidada             23
diferencia                                                     9
  de ellas, creadas para pruebas y nunca cerradas              6
  de ellas, de un equipo que se disolvió                       2
  de ellas, sin ningún dueño identificable                     1
```

Nueve cuentas fuera de todo control. La que no tenía dueño identificable contenía:

```
un almacén de objetos con 41 GB, accesible públicamente
contenido    volcados de una base de datos de 2023, con datos de clientes
tiempo expuesto                                          19 meses
accesos externos registrados                no se sabe: sin registro
```

Diecinueve meses. Y ninguna herramienta lo habría visto, porque la cuenta no estaba en el alcance de nada.

```
cuentas conocidas                          14             23
cuentas analizadas                          0             23
cuentas cerradas por no tener uso           —              6
cuentas con dueño asignado                 14             23
```

Los 12.400 hallazgos.

```
hallazgos en la primera ejecución completa                  12.410
por gravedad nominal
  crítica                                                    1.180
  alta                                                       3.940
  media y baja                                               7.290
```

Y el embudo por exposición, que es el que se usó:

```
accesibles desde internet                                       44
que tocan datos personales                                      61
que permiten obtener credenciales o escalar permisos            18
que afectan al registro de auditoría o a las copias              9
                                                              ■■■■■■■
prioridad inmediata (con solapamiento)                         103
```

Ciento tres frente a doce mil cuatrocientos. Y el resto pasó a línea base congelada:

```
línea base                                                  12.307
objetivo de reducción                          800 por trimestre
cifra publicada                                                sí
regla                                        ningún hallazgo NUEVO
```

La atribución, que bloqueó todo durante tres semanas.

```
hallazgos que se pudieron asignar a un equipo                 41 %
recursos sin etiqueta de dueño                               4.180
recursos con etiqueta de un equipo inexistente                 610
recursos que no aparecían en ninguna declaración              1.940
```

Y el orden que se siguió fue el del apartado segundo: primero poder repartir, después repartir.

```
semana 1   política del proveedor: no se puede crear un recurso
           sin etiquetas de equipo y entorno
           → resuelve el futuro, no el pasado
semana 2   campaña de etiquetado por cuenta, usando el registro
           de auditoría para saber quién creó cada cosa
semana 3   recursos sin dueño tras la campaña: 118
           → se dieron 30 días; lo que siguiera sin dueño se apagaría
           → al vencer, 11 recursos apagados; 2 reclamaciones,
             ambas resueltas en el día
```

```
hallazgos asignables                          41 %           98 %
recursos sin etiqueta de dueño               4.180           118 → 0
recursos fuera de la infraestructura declarada 1.940          310
```

Los 310 restantes son recursos creados por otros servicios —discos de un clúster, reglas de red de un balanceador— y se documentaron como tales.

Las reglas propias, y la que estaba rota.

```
reglas del catálogo por defecto                                241
reglas propias escritas                                         18
reglas con prueba negativa                                  259 de 259
```

Y al escribir las pruebas negativas apareció lo esperable:

```
reglas que no detectaban lo que decían detectar                  7
  de ellas, del catálogo por defecto                             4
  de ellas, propias                                              3
una de ellas    «ninguna base de producción admite conexiones desde dev»
                no comprobaba las reglas heredadas del grupo padre
                → llevaba 4 meses dando cero hallazgos
                → y había 2 bases que la incumplían
```

Es la ley 13 en su forma de regla rota: cero hallazgos parecía cumplimiento perfecto.

La corrección automática, y el incidente.

La primera regla automatizada fue «cerrar el acceso público de cualquier almacén de objetos». Parecía obviamente segura.

```
14:20  se activa en las 23 cuentas a la vez
14:21  se cierran 61 almacenes
14:26  el sitio web público deja de mostrar imágenes de producto
causa  4 de los 61 servían contenido estático público a propósito
14:44  revertido a mano
```

```
duración                                                    24 min
```

Y el diagnóstico es el del apartado cuarto: se aplicó a todas las cuentas a la vez, sin simulación previa y sin excepción declarada.

```
modo simulación previo                          no            sí, 7 días
alcance inicial                           23 cuentas      1 cuenta de dev
escalonado                                      no          dev → pre → pro
excepciones declaradas junto al recurso         no          4, con motivo
contador y límite                               no       20 actuaciones/día
registro de actuaciones                         no             sí
incidentes por corrección automática             1              0
```

Y el contador reveló algo a las dos semanas:

```
actuaciones de la regla «falta etiqueta de equipo»            188/día
causa    un servicio creaba recursos temporales sin etiquetas
         y la corrección los etiquetaba uno a uno
→ la corrección automática estaba TAPANDO un defecto
→ ley 19: se corrigió el servicio, y las actuaciones bajaron a 0-2/día
```

La corrección en el origen frente al recurso.

```
primeros 30 hallazgos corregidos tocando el recurso
  revertidos por el bucle de reconciliación en menos de 5 min      22
  supervivientes (recursos no gobernados por el bucle)              8
```

Veintidós de treinta correcciones duraron menos de cinco minutos. Se cambió el mecanismo:

```
                                                          al repositorio
correcciones que sobreviven                 8 de 30          30 de 30
tiempo medio hasta aplicar                   5 min            4 h
queda registro de por qué                      no              sí
pelea entre automatismos                       sí              no
```

Cuatro horas en lugar de cinco minutos, y las correcciones se quedan.

A los seis meses.

```
cuentas analizadas                            0 de 23        23 de 23
almacenes públicos con datos                      1              0
hallazgos totales                            12.410          6.900
hallazgos de prioridad inmediata               103              4
antigüedad del más viejo entre los expuestos  19 meses        5 días
hallazgos asignables a un equipo               41 %           98 %
recursos sin dueño                            4.180             0
reglas con prueba negativa                    0 de 241      259 de 259
reglas rotas encontradas                          —             7
correcciones automáticas con contador             0        11 de 11
incidentes por corrección automática              1              0
correcciones que sobreviven al bucle          8 de 30        30 de 30
```

La lección que esta clase traslada a la parte 11: el hallazgo más grave —un almacén público con volcados de clientes durante diecinueve meses— estaba en una cuenta que no aparecía en ningún inventario, así que ninguna herramienta lo habría encontrado por muy bien configurada que estuviera. Y lo que bloqueó el programa durante tres semanas no fue la seguridad: fue no saber de quién era cada recurso, que es exactamente lo que la hipótesis de la clase 132 predijo como problema central de esta parte.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-11-security-governance-finops/139-cspm-postura-policy-as-code-y-remediacion/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 guardrail-policy 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 guardrail-policy 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 herramienta dice que todo está bien y hay recursos expuestos	La cobertura no incluye todas las cuentas y proyectos	Contrasta el inventario con la facturación consolidada y analiza todo lo que exista, no solo lo documentado.
Miles de hallazgos y ninguno se corrige	Ley 15, y además no se sabe a quién asignarlos	Congela la línea base, bloquea solo lo nuevo, prioriza por exposición y resuelve la atribución antes de repartir trabajo.
Una regla lleva meses sin producir hallazgos y parece cumplimiento	La regla está mal escrita y no detecta lo que dice detectar	Prueba negativa para cada regla: crea a propósito un recurso que la incumpla y comprueba que salta.
Las correcciones desaparecen a los pocos minutos	Se modificó el recurso y el bucle de reconciliación lo revirtió	Corrige la declaración en el repositorio con un cambio propuesto; si el recurso no está declarado, impórtalo o elimínalo.
Una corrección automática provoca una caída	Se aplicó a todo a la vez, sin simulación, sin escalonar y sin excepciones	Simula primero, escalona por entorno, declara excepciones junto al recurso y limita a lo que no puede cortar tráfico.
Una corrección automática actúa cientos de veces al día	Ley 19: está tapando un defecto que genera recursos mal configurados	Contador y límite en toda corrección automática; investiga la causa cuando la cifra sube.

■ 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 momentos de aplicar un control y qué criterio decide entre ellos?
2. ¿Por qué la detección continua hace falta aunque todo se despliegue declarado?

3. ¿Por qué un hallazgo sin dueño no es una tarea, y cómo se resuelve la atribución?
4. ¿Qué separa una regla de un deseo?
5. ¿Por qué la corrección debe hacerse en el repositorio y no en el recurso?

Referencias

- AWS (2025). Service control policies and organization-level guardrails — controles preventivos que no se rodean. <https://docs.aws.amazon.com/organizations/latest/userguide/orgsmanagepoliciesscps.html>
- Google Cloud (2025). Organization policy constraints — restricciones aplicadas a toda la jerarquía. <https://cloud.google.com/resource-manager/docs/organization-policy/overview>
- Azure (2025). Azure Policy: effects, exemptions and remediation tasks — detección, excepciones y corrección. <https://learn.microsoft.com/azure/governance/policy/concepts/effects>
- Open Policy Agent (2025). Policy testing — pruebas de reglas como código. <https://www.openpolicyagent.org/docs/latest/policy-testing/>
- CIS (2025). Benchmarks for cloud providers — catálogos de reglas de configuración por defecto. <https://www.cisecurity.org/cis-benchmarks>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 11 en PDF · Manual integral

Anterior	Índice	Siguiente
← 138 · Vulnerabilidades, imágenes y cadena de suministro	Parte 11 · Programa	140 · Threat modeling con STRIDE y attack paths →

Evaluación — 139 CSPM, postura, policy as code y remediación

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 guardrail-policy 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.

140 — Threat modeling con STRIDE y attack paths

Parte: 11 — Seguridad, gobierno, cumplimiento y FinOps Nivel: avanzado · Horas estimadas: 4 Laboratorio: security · Estado: EXECUTABLECORE

Propósito

Añadir lo que ninguna herramienta de las clases anteriores puede hacer: preguntarse qué intentaría alguien contra este sistema en concreto. La clase da una estructura que evita que la conversación dependa de la imaginación de quien esté en la sala —una lista de comprobación aplicada a cada frontera de confianza— y le suma el razonamiento que de verdad refleja cómo ocurren los ataques: seguir cadenas, porque tres hallazgos de gravedad baja encadenados producen un compromiso grave, y la gravedad no se suma.

Resultados de aprendizaje

Al finalizar podrás:

1. Dibujar flujos de datos y marcar las fronteras de confianza, que es donde está casi todo.
2. Aplicar una lista de comprobación de amenazas a cada elemento y cada flujo.
3. Construir cadenas de ataque desde cada punto de entrada hasta lo que importa.
4. Priorizar por existencia y longitud de camino, no por gravedad individual.
5. Convertir el ejercicio en cambios con dueño y en detecciones nuevas.

Conceptos centrales

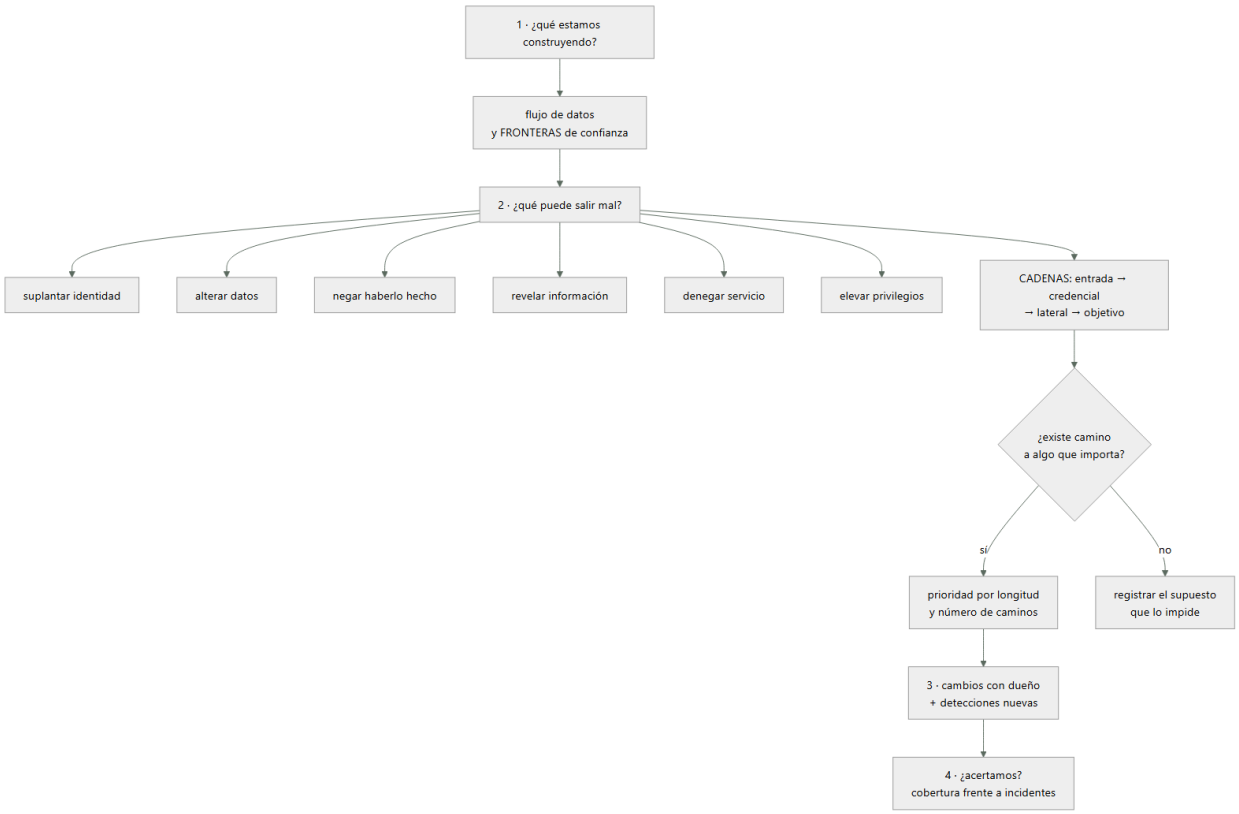
Concepto	Comprensión verificable
frontera de confianza	Punto donde el dato o el control pasa de un dominio a otro con distinto nivel de confianza. Es donde se concentran las vulnerabilidades.
lista de comprobación de amenazas	Conjunto fijo de categorías que se aplica a cada elemento para no depender de que alguien se acuerde.
cadena de ataque	Secuencia de pasos desde un punto de entrada hasta un objetivo. Es como ocurren los compromisos reales.
composición de gravedad	Que varios hallazgos leves encadenados produzcan uno grave. La gravedad individual no lo refleja.
supuesto	Afirmación de la que depende la seguridad del diseño y que se da por cierta. Cada uno es un riesgo si deja de cumplirse.
cobertura del modelo	Proporción de incidentes reales que el modelo había previsto. Es la única forma de saber si el ejercicio sirve.

Modelo mental

Gobernar no significa aprobar cada cambio: significa codificar límites, evidencia y responsabilidades para que los equipos puedan avanzar con seguridad.

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. Cuatro preguntas y un dibujo

El ejercicio entero cabe en cuatro preguntas, y conviene no complicarlo más:

```
1. ¿QUÉ ESTAMOS CONSTRUYENDO?
2. ¿QUÉ PUEDE SALIR MAL?
3. ¿QUÉ VAMOS A HACER AL RESPECTO?
4. ¿LO HICIMOS BIEN?
```

La cuarta es la que casi nadie responde, y la única que dice si el ejercicio sirve para algo.

Cuándo hacerlo, porque hacerlo para todo es la forma de no hacerlo para nada:

```
sí   un sistema nuevo
     una frontera de confianza nueva: un integrador, un canal, un tercero
     algo que toque dinero, datos personales o autenticación
     un cambio de arquitectura importante
no   cada cambio pequeño
     lo que no cruza ninguna frontera nueva
```

El dibujo es lo que hace posible la segunda pregunta, y su valor no está en las cajas:

```
se dibujan     los flujos de datos entre elementos
               dónde se almacena qué
               quién ejecuta cada parte
se marcan      las FRONTERAS DE CONFIANZA
```

Y las fronteras habituales, que son las que hay que buscar:

```
internet ←→ tu sistema
un cliente ←→ tu API                          clase 118
un servicio ←→ otro servicio                  clase 135
la aplicación ←→ la base de datos             clase 109
tu sistema ←→ un tercero                      clase 137
un entorno inferior ←→ producción             clase 133
una persona ←→ una herramienta interna
el proceso de construcción ←→ producción      clase 098
```

Y la razón de centrarse en ellas:

```
casi todas las vulnerabilidades viven en una frontera
```

```
porque es donde cambia quién controla el dato o la ejecución
```

Los supuestos, que es la parte del dibujo que más rinde y que casi nunca se escribe:

```
«el tráfico interno está autenticado»
«este servicio solo lo llama la puerta de entrada»
«los datos de este campo ya vienen validados»
«el proveedor valida la firma del aviso que nos envía»
```

Cada supuesto es un riesgo si deja de cumplirse, y escribirlos convierte una suposición en algo comprobable. En la práctica, una parte de los hallazgos del ejercicio consiste en descubrir que un supuesto nunca fue cierto.

2. Una lista para no depender de la inspiración

«¿Qué puede salir mal?» respondido de memoria produce lo que a cada uno le preocupa. Una lista fija fuerza cobertura. Las seis categorías clásicas, con dónde se resuelve cada una en este programa:

```
SUPLANTAR IDENTIDAD
  ¿cómo sabe este elemento quién le llama?
  → identidad de carga, autenticación mutua      clases 133, 135, 137

ALTERAR DATOS
  ¿puede alguien modificar el dato en tránsito, en reposo o el artefacto?
  → cifrado en tránsito, firma, validación de entrada   clases 067, 136

NEGAR HABERLO HECHO
  ¿queda constancia de quién hizo qué, y es inalterable?
  → registro de auditoría inmutable y sin identidades compartidas
                                                 clases 134, 112

REVELAR INFORMACIÓN
  ¿qué información sale por errores, registros, respuestas o salida de red?
  → depuración de registros, control de salida    clases 122, 135

DENEGAR SERVICIO
  ¿qué operación es cara y quién puede pedirla muchas veces?
  → límites por cliente, descarte, cuotas         clases 118, 130

ELEVAR PRIVILEGIOS
  ¿desde aquí se puede conseguir más de lo que corresponde?
  → privilegio mínimo, fronteras de permisos      clase 134
```

Y la forma práctica de aplicarla, que evita las sesiones interminables:

```
para cada FLUJO que cruza una frontera, se recorren las seis
se anota solo lo que no esté ya resuelto y sea plausible
y se pasa al siguiente
```

Y tres preguntas transversales que producen más hallazgos que las seis categorías juntas:

```
¿qué pasa si este elemento está comprometido?
¿qué pasa si el dato de entrada es malicioso en vez de erróneo?
¿qué pasa si esta llamada se repite mil veces o llega dos veces?
```

La tercera enlaza directamente con la clase 116: repetir es un caso de uso del atacante, no solo un accidente de la red.

Y una advertencia sobre la dinámica de la sesión, que decide su calidad:

```
en la sala hacen falta quienes construyeron el sistema
no es una auditoría: quien la dirige hace preguntas, no acusaciones
se acota a 90 minutos y un ámbito concreto
y se sale con una lista de cambios, no con un documento
```

3. Pensar en cadenas

La lista anterior enumera amenazas por elemento. Los ataques reales no ocurren por elemento: ocurren siguiendo una cadena.

```
entrada  →  credencial  →  movimiento lateral  →  objetivo
```

Y de ahí sale la observación central de este apartado:

```
LA GRAVEDAD NO SE SUMA

tres hallazgos individuales de gravedad baja
encadenados producen un compromiso grave
```

```
y cada uno, por separado, estaba correctamente clasificado como bajo
```

Por eso hay que recorrer los caminos, no la lista. El método es el ejercicio de la clase 133, hecho con detalle:

```
1. enumerar los puntos de entrada plausibles
2. desde cada uno, ¿qué se puede obtener?
3. con eso, ¿a qué se puede llegar?
4. repetir hasta que no haya paso nuevo
5. marcar los caminos que terminan en algo que importa
```

Y qué es «algo que importa» hay que escribirlo antes, o el ejercicio no termina:

```
datos personales de clientes
la capacidad de mover dinero
la capacidad de desplegar código               clase 098
el registro de auditoría
las copias de seguridad
las claves                                     clase 136
```

Y la priorización, que es más útil que multiplicar probabilidad por impacto:

```
¿EXISTE un camino?                  sí o no; ya es una respuesta
¿cuántos pasos tiene?               cuanto más corto, peor
¿cuántos caminos distintos llevan
  al mismo objetivo?                muchos caminos = un solo control
                                    no lo va a arreglar
¿qué paso aparece en más caminos?   ese es el que hay que cortar primero
```

La última pregunta es la que da el mayor rendimiento: suele haber un paso que aparece en la mayoría de los caminos, y cortarlo elimina familias enteras.

Y los pasos que aparecen una y otra vez en este programa:

```
una credencial permite obtener otra             clase 133
un entorno inferior alcanza producción          clase 133
una identidad puede modificar sus propios permisos  clase 134
un servicio comprometido alcanza a los demás    clase 135
la canalización puede desplegar sin revisión    clase 098
```

Y una nota sobre las herramientas: hay productos que calculan estos caminos automáticamente sobre la configuración real. Sirven, y no sustituyen al ejercicio, porque no conocen la lógica de negocio ni los supuestos: una herramienta no sabe que ese campo lo rellena un tercero.

4. Que sirva para algo

La tercera pregunta —qué vamos a hacer— tiene cuatro respuestas posibles, y todas son legítimas menos la ausencia:

```
CORREGIR       cambiar el diseño o añadir un control
TRANSFERIR     que lo asuma un tercero, con contrato
ACEPTAR        con nombre, motivo y fecha de revisión
ELIMINAR       quitar la funcionalidad o el dato que crea el riesgo
```

La cuarta se olvida y suele ser la mejor: el dato que no se guarda no se puede filtrar. Buena parte de los hallazgos sobre datos personales se resuelven no recogiéndolos, no guardándolos tanto tiempo o guardando una referencia en lugar del valor.

Y el resultado del ejercicio tiene que ser accionable, no documental:

```
mal   un documento de veinte páginas que nadie vuelve a abrir
bien  una lista de cambios con dueño y fecha
      + detecciones nuevas para lo que no se pueda prevenir
      + supuestos escritos, con quién los verifica
      + un diagrama de una página que se mantiene
```

La segunda línea enlaza con la parte 10: lo que no se puede impedir, se detecta, y ese es el origen de alertas que ninguna otra actividad produce.

La cuarta pregunta —¿lo hicimos bien?— se responde con el tiempo y con datos:

```
incidentes ocurridos que el modelo había previsto
incidentes ocurridos que NO estaban en el modelo   ← lo importante
supuestos que resultaron falsos
cambios propuestos que se completaron
```

Y la segunda línea es la que mejora el ejercicio siguiente: cada incidente no previsto señala una categoría que la lista no cubrió o una frontera que no se dibujó.

Y cuándo repetirlo:

```
al cambiar la arquitectura
al añadir una frontera nueva
al empezar a tratar una categoría de dato nueva
tras un incidente que no estaba previsto
y si nada de lo anterior ocurre, una vez al año
```

Y la lista de comprobación de la clase:

- el ámbito está acotado y hay un dibujo de una página
- las fronteras de confianza están marcadas explícitamente
- los supuestos están escritos y asignados a alguien que los verifique
- se ha recorrido la lista de categorías por cada flujo que cruza frontera
- se han preguntado las tres transversales, incluida la de repetir
- se han construido cadenas desde cada punto de entrada
- está escrito qué se considera «algo que importa»
- se ha identificado el paso que aparece en más caminos
- cada hallazgo tiene una de las cuatro respuestas, no ninguna
- hay detecciones nuevas para lo que no se puede prevenir
- se mide la cobertura frente a los incidentes reales

Y el cierre que enlaza con la clase siguiente: varios de los hallazgos de esta clase se refieren a datos personales, a dónde pueden estar y a quién puede verlos. Esas restricciones no las decide la ingeniería, y demostrar que se cumplen es un trabajo propio, que es la materia de la clase 141.

Ejemplo trabajado

CloudShop modela el flujo de pago antes de integrar un segundo proveedor. La sesión dura noventa minutos y produce nueve hallazgos; el ejercicio de cadenas, hecho aparte, produce uno más grave que los nueve juntos.

El dibujo y sus fronteras.

```
fronteras identificadas                                          8
  navegador ←→ puerta de entrada
  puerta ←→ servicio de pedidos
  pedidos ←→ base de datos
  pedidos ←→ cola de eventos
  motor durable ←→ proveedor de pago A
  motor durable ←→ proveedor de pago B (nueva)
  proveedor B ←→ nuestro receptor de avisos (nueva)
  herramienta interna de soporte ←→ datos de pedidos
```

Las dos nuevas son las que motivaban la sesión. La octava no estaba en el diagrama original y apareció al preguntar quién más toca estos datos.

Los supuestos escritos, y los dos que eran falsos.

```
1. «el tráfico entre servicios está autenticado»          cierto (clase 135)
2. «el receptor de avisos valida la firma del proveedor»  FALSO
3. «los importes vienen validados por el proveedor»       parcialmente falso
4. «solo soporte accede a la herramienta interna»         FALSO: 11 personas
                                                          con acceso, 4 sin
                                                          necesitarlo
5. «un aviso repetido no produce efecto»                  cierto (clase 116)
```

El supuesto 2 es el hallazgo más directo de la sesión:

```
el receptor de avisos aceptaba cualquier petición con el formato correcto
→ cualquiera que conociera la URL podía marcar un pedido como pagado
tiempo que llevaba así                  desde la integración: 14 meses
explotado                               no, según los registros
```

La lista aplicada a las dos fronteras nuevas.

```
SUPLANTAR
  el receptor no verifica quién envía        ← hallazgo, arriba
  nosotros hacia el proveedor B: clave estática compartida por 3 cargas
    → una identidad por carga                 clase 137

ALTERAR
```

```
  el aviso viaja firmado y no se comprueba   ← el mismo hallazgo
  el importe se toma del aviso y no se contrasta con el pedido
    → hallazgo: contrastar siempre contra lo esperado

NEGAR HABERLO HECHO
  la herramienta de soporte permite cambiar el estado de un pedido
  sin registrar quién                       ← hallazgo

REVELAR INFORMACIÓN
  el error del receptor devolvía el mensaje del proveedor completo
    → incluía identificadores internos       ← hallazgo
  la herramienta de soporte muestra la clave de integración (clase 137)

DENEGAR SERVICIO
  el receptor no tiene límite de ritmo       ← hallazgo
  cada aviso dispara una consulta cara al motor durable

ELEVAR PRIVILEGIOS
  la identidad del receptor puede escribir en la cola de eventos
  de CUALQUIER tipo, no solo de pago         ← hallazgo
```

Nueve hallazgos, todos con dueño y fecha. Y las tres preguntas transversales añadieron dos más:

```
«¿y si el aviso llega dos veces?»       cubierto (clase 116)
«¿y si llega mil veces?»                no cubierto → límite añadido
«¿y si el dato es malicioso?»           el campo de referencia se
                                        concatenaba en una consulta
                                        → hallazgo grave
```

El ejercicio de cadenas, hecho aparte.

Se definió primero qué importa:

```
datos personales de clientes · mover dinero · desplegar código ·
registro de auditoría · copias · claves
```

Y se recorrieron los caminos desde cada punto de entrada. El más corto hasta «mover dinero»:

```
paso 1  la herramienta interna de soporte está expuesta a la red
        corporativa, sin autenticación de segundo factor
        gravedad individual: MEDIA

paso 2  la herramienta muestra la clave de integración del proveedor B
        gravedad individual: BAJA (era «solo lectura de configuración»)

paso 3  esa clave permite emitir devoluciones en el proveedor B
        gravedad individual: BAJA (la clave se consideraba «de consulta»)

→ CADENA: acceso corporativo → clave → devoluciones a cuentas arbitrarias
→ gravedad de la cadena: CRÍTICA
```

Tres hallazgos clasificados como medio, bajo y bajo. Ninguna herramienta de las clases 138 o 139 los habría relacionado, y los tres estaban correctamente clasificados por separado.

Y el paso que aparecía en más caminos:

```
caminos construidos hasta objetivos que importan                14
caminos que pasaban por la herramienta de soporte                9
caminos que pasaban por «una credencial da otra»                11

cortar el acceso amplio a la herramienta de soporte     elimina 9 caminos
ocultar credenciales en las herramientas internas       elimina 11
→ dos cambios eliminan 12 de los 14 caminos
```

Lo que se hizo con los hallazgos.

```
CORREGIR    validación de firma en el receptor
            contraste del importe contra el pedido
            límite de ritmo en el receptor
            identidad por carga hacia el proveedor B
            permisos del receptor acotados a un tipo de evento
            consulta parametrizada en el campo de referencia
            segundo factor y permisos mínimos en la herramienta de soporte
            ocultación de credenciales en herramientas internas

ELIMINAR    el mensaje del proveedor deja de devolverse al cliente
```

```
            la herramienta de soporte deja de mostrar la configuración
              completa: solo los campos que soporte necesita

ACEPTAR     el proveedor A no admite identidad federada
            → clave estática rotada cada 30 días, con nombre y revisión
              semestral

DETECTAR    alerta por avisos con firma inválida
            alerta por devoluciones fuera del horario habitual
            alerta por acceso a la herramienta de soporte desde fuera
              del horario
```

Las cuatro detecciones no existían y ninguna otra actividad las habría producido.

La cuarta pregunta, respondida a los doce meses.

```
incidentes de seguridad en el periodo                              3
  previstos en el modelo                                           2
    → un intento de aviso falsificado, bloqueado por la validación
    → un pico de avisos, contenido por el límite
  NO previstos                                                     1
    → un empleado con acceso legítimo exportó datos de clientes
      antes de irse

supuestos verificados                                        5 de 5
supuestos que resultaron falsos                                  2
cambios propuestos completados                              14 de 15
```

Y el incidente no previsto señaló una categoría que la lista no cubría bien: el uso legítimo con intención maliciosa. Se añadió una pregunta a la lista para las revisiones siguientes:

```
«¿qué puede hacer con esto alguien que TIENE acceso legítimo
 y quiere hacer daño, y cómo nos enteraríamos?»
```

Y esa pregunta, aplicada al resto del sistema, produjo once hallazgos más y tres detecciones nuevas por volumen de exportación.

El resultado del ejercicio.

```
fronteras de confianza documentadas              0              8
supuestos escritos                               0              5
de ellos, falsos al comprobarlos                 —              2
hallazgos del ejercicio                          —             11
cadenas hasta objetivos críticos                14              2
pasos que aparecían en más de la mitad
de los caminos                                   2              0
detecciones nuevas                               0              7
incidentes previstos por el modelo               —          2 de 3
```

La lección que esta clase traslada a la parte 11: la sesión estructurada encontró nueve problemas reales, uno de ellos con catorce meses de antigüedad —cualquiera que conociera una dirección podía marcar pedidos como pagados—. Y el hallazgo más grave no lo produjo la lista de amenazas, sino seguir una cadena de tres hallazgos que estaban correctamente clasificados como medio, bajo y bajo. Ninguna herramienta los relacionó, y dos cambios eliminaron doce de los catorce caminos: cortar el paso que se repetía en casi todos.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-11-security-governance-finops/140-threat-modeling-con-stride-y-attack-paths/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 modelo-amenazas 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 modelo-amenazas 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 sesión produce las amenazas que preocupan a quien más habla	Se responde de memoria en vez de recorrer una lista fija	Aplica las seis categorías a cada flujo que cruza una frontera, más las tres preguntas transversales.
El ejercicio termina en un documento que nadie vuelve a abrir	El resultado no son cambios con dueño	Sal con una lista de cambios con dueño y fecha, detecciones nuevas y un diagrama de una página que se mantenga.
Tres hallazgos leves acaban produciendo un compromiso grave	La gravedad individual no compone; nadie recorrió las cadenas	Construye caminos desde cada punto de entrada hasta lo que importa y corta el paso que aparece en más caminos.
La seguridad del diseño depende de algo que nunca fue cierto	Los supuestos no se escribieron ni se verificaron	Escribe cada supuesto, asígnalo a alguien y compruébalo; suelen ser una fuente directa de hallazgos.
Se modela todo y no se modela nada a tiempo	Ámbito sin acotar	Reserva el ejercicio para sistemas nuevos, fronteras nuevas y lo que toque dinero, datos personales o autenticación.
Ocurren incidentes que el modelo no contemplaba y nadie revisa el método	No se responde la cuarta pregunta	Mide cuántos incidentes estaban previstos y usa los que no lo estaban para añadir categorías o fronteras.

■ 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 valor del diagrama está en las fronteras y no en los elementos?
2. ¿Qué aporta una lista fija de categorías frente a preguntar qué puede salir mal?
3. ¿Por qué la gravedad individual no basta y qué se hace en su lugar?

4. ¿Qué cuatro respuestas puede tener un hallazgo, y cuál se olvida más?
5. ¿Cómo se sabe si el ejercicio sirvió para algo?

Referencias

- Shostack, A. (2014). Threat Modeling: Designing for Security — las cuatro preguntas y la aplicación de la lista por elemento. <https://shostack.org/books/threat-modeling-book>
- Microsoft (2025). STRIDE threat categories — definición de las seis categorías y su uso. <https://learn.microsoft.com/azure/security/develop/threat-modeling-tool-threats>
- OWASP (2025). Threat modeling process — alcance, dinámica de la sesión y resultados accionables. <https://owasp.org/www-community/ThreatModelingProcess>
- MITRE (2025). ATT&CK: tactics and techniques — pasos reales de una cadena de ataque. <https://attack.mitre.org/>
- Threat Modeling Manifesto (2025). Values and principles — por qué el resultado importa más que el documento. <https://www.threatmodelingmanifesto.org/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 11 en PDF · Manual integral

Anterior	Índice	Siguiente
← 139 · CSPM, postura, policy as code y remediación	Parte 11 · Programa	141 · Cumplimiento, residencia, privacidad y evidencia →

Evaluación — 140 Threat modeling con STRIDE y attack paths

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 modelo-amenazas 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.

141 — Cumplimiento, residencia, privacidad y evidencia

Parte: 11 — Seguridad, gobierno, cumplimiento y FinOps Nivel: avanzado · Horas estimadas: 4 Laboratorio: compliance · Estado: EXECUTABLECORE

Propósito

Demostrar a alguien que no estaba delante que los controles existen y funcionan. La clase separa dos cosas que se confunden —tener seguridad y poder demostrarla— y defiende que la evidencia debe ser un subproducto continuo de cómo se opera, no una campaña anual de capturas de pantalla. Después trata los tres asuntos que más incumplimientos producen en la práctica: dónde acaban de verdad los datos, cómo se borra lo que hay que borrar cuando el sistema está lleno de registros inmutables, y qué ocurre cuando quien incumple es un proveedor.

Resultados de aprendizaje

Al finalizar podrás:

1. Distinguir tener un control de poder demostrarlo, y cubrir las dos cosas.
2. Generar evidencia como subproducto de la operación normal.
3. Clasificar los datos en el momento de recogerlos.
4. Comprobar dónde acaban los datos de verdad, incluidos registros y telemetría.
5. Borrar lo que hay que borrar en sistemas que no permiten borrar.

Conceptos centrales

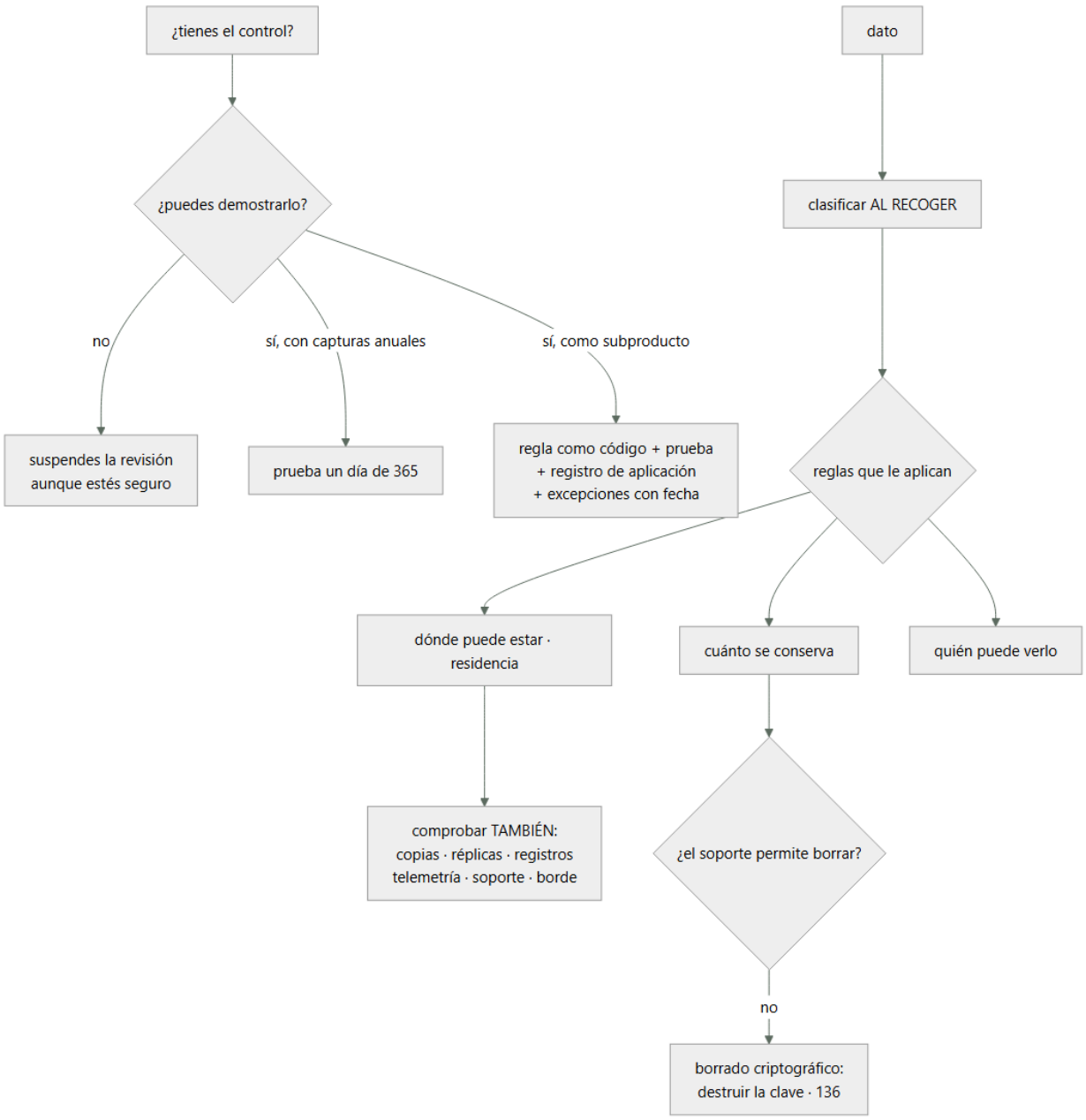
Concepto	Comprensión verificable
evidencia	Prueba verificable de que un control existió y se aplicó durante un periodo, no solo el día de la revisión.
clasificación	Etiqueta que dice qué tipo de dato es y qué reglas le aplican. Sin ella no se puede aplicar ninguna regla.
residencia	Requisito de que los datos permanezcan en una zona geográfica. Incluye copias, réplicas, registros, telemetría y acceso de soporte.
minimización	No recoger ni conservar lo que no hace falta. Es el control más eficaz y el único que elimina el riesgo en vez de gestionarlo.
borrado criptográfico	Destruir la clave que protege un dato para hacerlo ilegible, cuando el soporte no permite borrarlo.
encargado del tratamiento	Tercero que trata datos por cuenta tuya. Su incumplimiento es tu incidente.

Modelo mental

Gobernar no significa aprobar cada cambio: significa codificar límites, evidencia y responsabilidades para que los equipos puedan avanzar con seguridad.

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. Tener el control y poder demostrarlo

Son dos problemas distintos y fallar en cualquiera tiene consecuencias:

```
control sin evidencia    el sistema es seguro y la revisión se suspende
                         porque no hay forma de demostrarlo
evidencia sin control    la revisión se aprueba y ocurre una brecha
```

Y el segundo es más común de lo que parece, porque es más fácil producir un documento que un control. Es la ley 17 en su forma más pura: la certificación se convierte en el objetivo y el sistema no mejora.

La forma cara de generar evidencia, que es la habitual:

```
una vez al año alguien recorre los sistemas haciendo capturas de pantalla
cuesta semanas de varias personas
y demuestra el estado de UN día de trescientos sesenta y cinco
```

La forma que funciona es tratarla como subproducto de cómo ya se opera:

```
la regla existe como código, revisada y versionada       clase 139
tiene prueba negativa que demuestra que detecta          clase 139
```

```
su aplicación deja registro continuo
las excepciones están declaradas, con motivo y caducidad
y todo eso se consulta con una consulta, no con una captura
```

Y cuatro ejemplos de la traducción, con lo que ya existe en este programa:

```
«el acceso a producción está controlado»
  → registro de concesiones temporales, con quién y por qué   clase 134

«el código se revisa antes de desplegarse»
  → historial del repositorio de entorno                       clase 103

«los cambios en producción están autorizados»
  → cada despliegue enlaza con su confirmación y su aprobación clase 099

«las copias de seguridad se restauran periódicamente»
  → registro del ensayo trimestral, con su cronómetro          clase 088
```

Y la propiedad que hace válida la evidencia:

```
cubre un PERIODO, no un instante
no la puede modificar quien es objeto del control      clase 134
y se puede reproducir: la misma consulta da el mismo resultado
```

Y una consecuencia práctica que ahorra mucho: preparar una revisión deja de ser un proyecto. Si la evidencia son consultas guardadas, se ejecutan el día que haga falta.

2. Clasificar, o no se puede aplicar nada

Ninguna regla se puede aplicar a un dato que no está clasificado. Y la clasificación tiene dos formas de hacerse, con resultados muy distintos:

```
AL RECOGER      quien crea el campo declara qué es
                → se mantiene sola, porque forma parte del diseño
A POSTERIORI    alguien recorre las bases buscando datos personales
                → caduca en semanas y hay que repetirlo eternamente
```

Y los niveles, que conviene que sean pocos:

```
PÚBLICO         puede publicarse
INTERNO         no debería salir, y su filtración no es grave
CONFIDENCIAL    daño real si se filtra: contratos, precios, planes
PERSONAL        identifica a una persona; reglas legales aplicables
  y dentro de este, la categoría especial: salud, biometría, ideología
```

Y con cuatro etiquetas ya se puede decidir todo lo demás:

```
dónde puede estar          residencia y proveedores permitidos
cuánto se conserva         retención y borrado
quién puede verlo          permisos y herramientas internas
si puede salir en registros o telemetría          clases 122, 124
si puede ir a un entorno de pruebas               clase 104
```

Y el mecanismo que lo mantiene vivo, con lo que ya existe:

```
esquema del evento y del modelo con la clasificación por campo   clase 115
comprobación en la canalización: campo nuevo sin clasificar → falla
el catálogo dice qué sistemas tratan qué categorías              clase 095
```

Y la minimización, que es el control más eficaz y el que menos se considera:

```
el dato que no se recoge no se puede filtrar, ni hay que
  cifrarlo, ni borrarlo, ni justificarlo, ni notificarlo
```

Y las tres preguntas que la aplican, en orden:

```
¿hace falta recogerlo?
¿hace falta conservarlo tanto tiempo?
¿hace falta el valor, o basta una referencia o un resumen?
```

La tercera resuelve muchos casos: guardar un identificador en lugar del dato convierte un problema de privacidad en uno de control de acceso, que es mucho más fácil.

3. Dónde acaban los datos de verdad

«Los datos se quedan en esta región» es un requisito frecuente y casi siempre falso la primera vez que se comprueba. Lo que hay que revisar, y lo que suele fallar:

```
almacenamiento principal             casi siempre correcto
copias de seguridad                  a veces en otra región
réplicas de lectura                  a veces en otra región
REGISTROS                            ← falla muy a menudo
TELEMETRÍA                           ← falla muy a menudo
caché de borde                       ← contenido replicado por el mundo
servicio de correo o mensajería      destino fuera de la región
herramientas internas y de soporte   consola del proveedor de otro país
ACCESO DE SOPORTE DEL PROVEEDOR      personas fuera de la región
entornos de pruebas                  ← clase 104
```

Las tres marcadas son las que rompen el requisito casi siempre, y ninguna es obvia: un registro con el nombre y el correo de un cliente enviado a un sistema de observabilidad en otro continente es un traslado de datos personales.

Y los controles disponibles:

```
política del proveedor que restrinja las regiones utilizables   clase 139
selección explícita de región en cada servicio, incluida la telemetría
recolector propio que depure antes de enviar fuera              clase 124
y lista de permitidos de salida hacia destinos fuera de la región  clase 135
```

Y una nota honesta sobre los límites: los metadatos y el plano de control suelen procesarse fuera. Nombres de recursos, identificadores y facturación viajan aunque el contenido no. Conviene saberlo y decirlo, en vez de afirmar una pureza que no existe.

El borrado, que es el requisito técnicamente más difícil de esta parte, porque los sistemas de la parte 09 están llenos de cosas que no se borran:

```
registro conservado de eventos       inmutable por diseño     clase 114
lago de datos                        ficheros columnares      clase 112
copias de seguridad                  con retención larga
registros de auditoría               que no se pueden alterar clase 134
registros de aplicación              con su retención         clase 122
cachés                               y sus capas              clase 111
entornos de pruebas                  con subconjuntos         clase 104
```

Y las tres estrategias, en orden de preferencia:

```
1. NO PUBLICAR EL DATO donde no se pueda borrar
   el evento lleva identificadores; el dato personal vive en la base
   → es la decisión de la clase 115, tomada por un motivo legal

2. BORRADO CRIPTOGRÁFICO
   cada sujeto tiene su clave; el dato se cifra con ella
   borrar la clave hace ilegible el dato allí donde esté
   → resuelve copias, lago y registros conservados de una vez
   → y requiere lo de la clase 136: clave por sujeto, e inventario

3. REESCRITURA
   con formato de tabla se puede borrar por filas              clase 112
   → caro, y no alcanza a las copias antiguas
```

Y la segunda es la respuesta elegante y tiene un precio que hay que aceptar: la gestión de miles o millones de claves, y que perder una clave equivale a perder ese dato para siempre.

Y el procedimiento de una solicitud de borrado, que hay que tener escrito y probado:

```
localizar todos los sistemas que tienen datos del sujeto
  → sale del catálogo y de la clasificación, no de una investigación
ejecutar el borrado o la destrucción de clave
registrar qué se hizo, cuándo y sobre qué sistemas
y responder en el plazo legal
```

4. Terceros y revisiones

Los terceros que tratan datos por cuenta tuya son parte de tu superficie:

```
el proveedor de nube
el de correo o mensajería
el de análisis                                clase 133
el de observabilidad                          clase 121
el de pagos                                   clase 118
las herramientas de soporte y de atención
y los que estos a su vez usen
```

Y lo que hace falta de cada uno:

```
inventario, con qué categorías de datos trata cada uno
un contrato que fije qué puede hacer y qué no
qué certificaciones tiene, y de qué alcance
dónde trata los datos
sus propios subcontratistas
y cómo notifica una brecha, y en cuánto tiempo
```

Y el punto que se olvida: la brecha de tu proveedor es tu incidente. Se declara, se comunica y se gestiona con el proceso de la clase 127, aunque el fallo no sea tuyo.

Y una comprobación que casi nadie hace y que da sorpresas: revisar el alcance de la certificación que enseña un proveedor. Un certificado puede cubrir un producto y no el que tú usas, o una región distinta.

Las revisiones externas, con lo que las hace baratas:

```
lo que pide quien revisa
  la política escrita
  la evidencia de que se aplica, durante todo el periodo
  una muestra: «enséñame estos 25 casos concretos»
  y las excepciones, con su justificación

lo que las hace caras
  buscar la evidencia cuando la piden
  no poder demostrar el periodo, solo el día
  y las excepciones sin registrar, que aparecen en la muestra
```

Y el consejo práctico: preparar las consultas de evidencia antes de que las pidan, guardadas y reutilizables. Convierte semanas en horas y, además, permite comprobarlas durante el año.

Y la advertencia final, que es la ley 17 aplicada a esta materia:

```
si la medida pasa a ser «tenemos el certificado»,
se optimizará el certificado
y el sistema no tiene por qué mejorar
→ la contramedida es medir también lo de las clases 133 a 140:
  alcance por punto de entrada, permisos sin usar, hallazgos expuestos,
  cadenas hasta objetivos críticos
```

Y la lista de comprobación de la clase:

- cada control tiene evidencia consultable que cubre un periodo
- la evidencia no la puede alterar quien es objeto del control
- los datos se clasifican al recogerlos, y un campo sin clasificar falla
- está aplicada la minimización: recoger menos, conservar menos, referenciar
- se ha comprobado dónde acaban registros, telemetría, copias y caché de borde
- hay política del proveedor que restringe regiones
- está escrito qué metadatos salen igualmente
- el dato personal no se publica donde no se pueda borrar
- hay estrategia de borrado para los sistemas inmutables
- el procedimiento de solicitud de borrado está probado, no solo escrito
- existe inventario de terceros con las categorías que trata cada uno
- se revisa el alcance real de sus certificaciones
- las consultas de evidencia están preparadas de antemano

Y el cierre que enlaza con la clase siguiente: todo lo de esta parte cuesta dinero y compite con otras cosas. Y hay una disciplina que se ocupa exactamente de eso —de saber qué se gasta, de quién es y si merece la pena—, con un problema central que resultará ser el mismo de esta clase: la atribución. Es la materia de la clase 142.

Ejemplo trabajado

CloudShop se somete a su primera revisión externa y a la vez recibe una solicitud de borrado de un cliente. Los dos ejercicios revelan el mismo problema: nadie sabía dónde estaban los datos.

La primera revisión: dos semanas y media de trabajo.

```
controles a demostrar                                  61
evidencia disponible al empezar                   12 de 61
forma de la evidencia          capturas de pantalla y hojas de cálculo
personas dedicadas                                      4
duración de la preparación                        12 días
observaciones del revisor                              19
```

```
  de ellas, «el control existe y no se puede demostrar»         11
  de ellas, control ausente                                      8
```

Once de diecinueve observaciones eran de evidencia, no de seguridad. Y la más ilustrativa:

```
control      «solo personal autorizado accede a producción»
realidad     cierto desde la clase 134: concesiones temporales
evidencia    una captura del panel de permisos, del día de la revisión
observación  «no demuestra el periodo; podría haber cambiado ayer»
```

La reconstrucción como subproducto.

```
forma de la evidencia            capturas puntuales    consultas guardadas
controles con evidencia               12 de 61            58 de 61
periodo cubierto                        1 día             12 meses
días de preparación                        12                   1,5
personas dedicadas                          4                     1
```

Y los tres restantes se documentaron como carencias reales, no como problemas de evidencia.

Ejemplos de la traducción:

```
«acceso a producción controlado»
  → consulta sobre el registro de concesiones: quién, cuándo, aprobado
    por quién, cuánto duró. 418 registros en 6 meses.

«los cambios están autorizados»
  → consulta sobre el repositorio de entorno: cada despliegue con su
    confirmación y su aprobador.

«las copias se restauran»
  → registro de los 4 ensayos trimestrales, con duración medida.

«los permisos se revisan»
  → registro de retiradas automáticas por desuso: 6.840 en 6 meses.
    → esta convenció más que cualquier acta de reunión
```

La residencia: el requisito que no se cumplía.

Un cliente europeo exigía que sus datos no salieran del espacio europeo. La revisión punto por punto:

```
base de datos principal                eu-west           sí
copias de seguridad                    eu-west           sí
réplica de lectura                     eu-west           sí
registros de aplicación                us-east           NO
telemetría (trazas y métricas)         us-east           NO
caché de borde                         global            NO
correos transaccionales                us-east           NO
herramienta de atención al cliente     us-east           NO
entorno de pruebas con subconjunto     eu-west           sí (clase 104)
```

Cinco de nueve incumplían, y las tres primeras eran las de siempre. Y el contenido que salía:

```
registros            nombre, correo y dirección en la línea ancha
                     → la lista de permitidos de la clase 122 los
                       incluía por descuido
telemetría           identificador de cliente en atributos de traza
caché de borde       páginas de pedido con datos del cliente
correos              nombre y dirección, por definición
atención al cliente  todo
```

```
registros                                  us-east         eu-west
  y los campos personales                  incluidos       fuera de la
                                                           lista de permitidos
telemetría                                 us-east         eu-west, y el
                                                           identificador
                                                           pseudonimizado
caché de borde                             global      páginas de cliente
                                                       marcadas como no
                                                       cacheables en borde
correos                                    us-east      proveedor europeo
atención al cliente                        us-east      instancia europea
política que restringe regiones              no             sí, 4 regiones
```

Y lo que se documentó como límite honesto:

```
metadatos del proveedor —nombres de recursos, facturación,
```

```
identificadores— se procesan fuera
→ escrito en el registro de tratamientos, no omitido
```

La solicitud de borrado, y por qué el catálogo lo resolvió.

```
sistemas que contenían datos del sujeto, según el catálogo         9
sistemas encontrados al buscar de verdad                          14
diferencia                                                         5
  registro conservado de eventos                                   sí
  lago de datos                                                    sí
  copias de seguridad de 14 meses                                  sí
  sistema de atención al cliente                                   sí
  hoja de cálculo de un equipo de marketing                        sí
```

Y los cuatro primeros eran los inmutables. La primera solicitud se resolvió a mano:

```
tiempo empleado                                          9 días
sistemas donde no se pudo borrar de verdad                  3
respuesta al cliente        «borrado de los sistemas activos;
                             las copias caducan en 14 meses»
```

Y se rediseñó con las dos primeras estrategias del apartado tercero:

```
1. NO PUBLICAR
   los eventos pasaron a llevar solo identificadores          clase 115
   → el registro conservado dejó de contener datos personales
   → tiempo de adopción: 6 semanas, con convivencia de esquemas

2. BORRADO CRIPTOGRÁFICO
   una clave por cliente; los campos personales se cifran con ella
   destruir la clave los hace ilegibles en base, lago y copias
   → apoyado en la clave por inquilino de la clase 136
```

```
tiempo de respuesta                    9 días              40 min
sistemas con dato legible tras el borrado    3                  0
sistemas que hubo que buscar a mano          5                  0
coste operativo por solicitud           ~2 días persona      ~15 min
solicitudes atendidas en 12 meses            —                 34
```

Y el coste asumido, que hay que decir:

```
claves gestionadas                                      190.000
coste mensual del servicio de claves                +  180 €
riesgo nuevo    perder una clave equivale a perder ese dato
                → copias de la jerarquía de claves, con su propia custodia
```

Los terceros.

```
terceros que tratan datos, según el inventario inicial          6
terceros encontrados al revisar la facturación y la salida     14
  → 8 más, incluidos 3 con acceso a datos personales

con contrato de tratamiento firmado                       6 de 14 → 14 de 14
con alcance de certificación verificado                   0 de 14 → 14 de 14
  → 2 tenían certificación de un producto DISTINTO al que usábamos
con plazo de notificación de brecha por escrito           3 de 14 → 14 de 14
```

Y el proveedor de análisis de la clase 133 volvió a aparecer: seguía en el inventario de terceros aunque el contrato había terminado, que es lo mismo que descubrió el control de salida de la clase 135.

A los doce meses.

```
controles con evidencia consultable           12 de 61       58 de 61
periodo cubierto por la evidencia               1 día        12 meses
días de preparación de una revisión               12            1,5
observaciones de revisión                         19              3
campos clasificados al recoger                    no             sí
sistemas fuera de la región exigida            5 de 9          0 de 9
política que restringe regiones                   no             sí
tiempo de respuesta a una solicitud de borrado   9 días        40 min
sistemas con dato legible tras un borrado          3              0
terceros inventariados                          6 de 14       14 de 14
certificaciones con alcance verificado          0 de 14       14 de 14
```

La lección que esta clase traslada a la parte 11: once de las diecinueve observaciones de la primera revisión decían que el control existía y no se podía demostrar, y se resolvieron sin cambiar ningún control: solo convirtiendo la

evidencia en consultas sobre lo que la operación ya registraba. Y los dos problemas reales —cinco sistemas fuera de la región exigida y datos personales en cuatro sitios de los que no se pueden borrar— tenían el mismo origen que la clase 139: nadie sabía dónde estaba cada cosa.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-11-security-governance-finops/141-cumplimiento-residencia-privacidad-y-evidencia/lab.py
```

El laboratorio selecciona el motor de práctica compliance 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 matriz-controles 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 mapeado a evidencia y responsable. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye matriz-controles 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 control funciona y la revisión lo marca como no demostrado	La evidencia es una captura de un día, no un registro del periodo	Convierte la evidencia en consultas sobre lo que la operación ya registra: concesiones, despliegues, ensayos, retiradas.
Preparar una revisión ocupa semanas de varias personas	La evidencia se recolecta cuando la piden	Prepara las consultas de antemano, guardadas y reutilizables, y compruébalas durante el año.
Se afirma que los datos no salen de una región y sí salen	Nadie revisó registros, telemetría, caché de borde, correos ni herramientas de soporte	Revisa los nueve destinos, restringe regiones por política del proveedor y depura los campos personales antes de enviar telemetría.
Una solicitud de borrado no se puede cumplir del todo	El dato está en registros conservados, lago y copias, que no permiten borrar	No publiques datos personales donde no se puedan borrar y usa borrado criptográfico con clave por sujeto.
Aparecen sistemas con datos del cliente que no estaban en el inventario	La clasificación se hizo a posteriori y caducó	Clasifica al recoger, falla la canalización si hay un campo sin clasificar y mantén el catálogo con qué sistema trata qué categoría.
Se obtiene la certificación y el riesgo real no baja	Ley 17: la certificación se convirtió en el objetivo	Mide además alcance por punto de entrada, permisos sin usar, hallazgos expuestos y cadenas hasta objetivos críticos.

Seguridad, ética y costo

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

Preguntas de comprobación

1. ¿Qué diferencia hay entre tener un control y poder demostrarlo, y cómo falla cada lado?
2. ¿Qué propiedades debe tener una evidencia para ser válida?
3. ¿Qué destinos suelen incumplir un requisito de residencia?
4. ¿Cómo se borra un dato que está en un registro conservado y en copias antiguas?
5. ¿Por qué la brecha de un proveedor es tu incidente?

Referencias

- Unión Europea (2016). Reglamento general de protección de datos — bases de tratamiento, minimización y derechos. <https://eur-lex.europa.eu/eli/reg/2016/679/oj>
- NIST (2020). Privacy Framework — clasificación, minimización y controles de privacidad. <https://www.nist.gov/privacy-framework>
- ISO/IEC (2022). 27001 e 27701: requisitos y evidencia — qué se audita y cómo se demuestra. <https://www.iso.org/standard/27001>
- Google Cloud (2025). Data residency and sovereignty controls — límites reales, metadatos y plano de control. <https://cloud.google.com/architecture/framework/security/data-residency-sovereignty>
- ENISA (2025). Crypto-shredding and data deletion in immutable systems — borrado criptográfico y sus límites. <https://www.enisa.europa.eu/publications>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Evaluación — 141 Cumplimiento, residencia, privacidad y evidencia

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 matriz-controles 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.

142 — FinOps: showback, chargeback, budgets y anomalías

Parte: 11 — Seguridad, gobierno, cumplimiento y FinOps Nivel: avanzado · Horas estimadas: 4 Laboratorio: finops · Estado: EXECUTABLECORE

Propósito

Gobernar un gasto que no decide nadie en concreto: la factura de la nube es la suma de cientos de decisiones pequeñas tomadas por personas que no ven el precio de lo que eligen. Por eso la clase no trata de negociar con el proveedor, sino de poner el coste al lado de la decisión. Y empieza por el problema que la hipótesis de la clase 132 predijo como central de toda esta parte: sin saber de quién es cada gasto, no se puede hacer nada con él.

Resultados de aprendizaje

Al finalizar podrás:

1. Atribuir cada gasto a un equipo y a un servicio, incluidos los compartidos.
2. Elegir entre mostrar el gasto y cobrarlo, sabiendo qué provoca cada uno.
3. Medir el coste por unidad de negocio, que es lo único comparable en el tiempo.
4. Detectar desviaciones sin ahogarse en falsas alarmas.
5. Reconocer el desperdicio que es un fallo de atribución y no de eficiencia.

Conceptos centrales

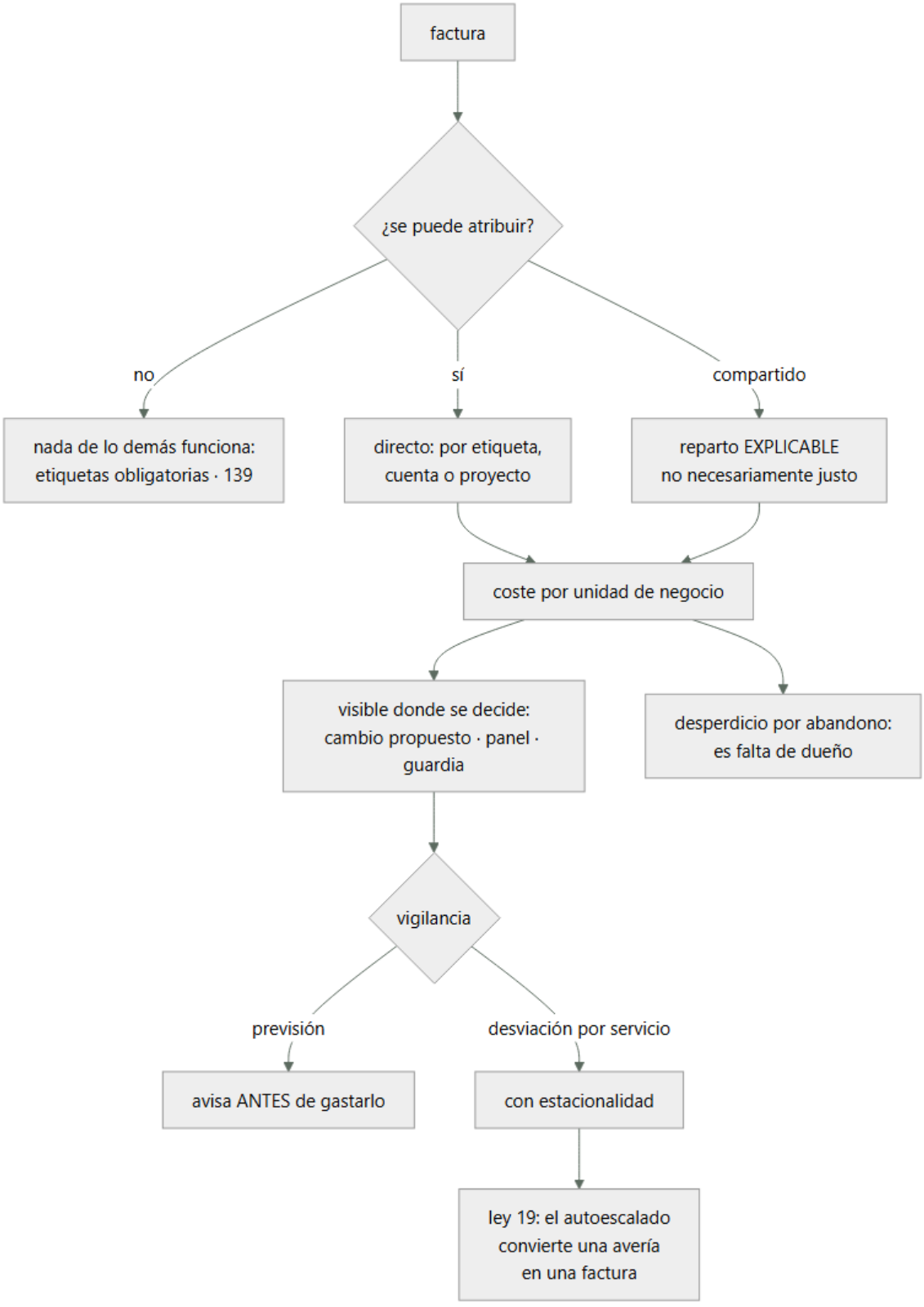
Concepto	Comprensión verificable
atribución	Saber a qué equipo y a qué servicio corresponde cada euro. Es el requisito previo de todo lo demás.
reparto de compartidos	Método para asignar lo que no es de nadie en concreto: red, clúster, plataforma, observabilidad.
mostrar frente a cobrar	Enseñar a cada equipo su gasto, o cargárselo en su presupuesto. Lo segundo cambia el comportamiento y también lo distorsiona.
coste unitario	Coste por pedido, por cliente o por petición. Es lo único que se puede comparar cuando el negocio crece.
presupuesto y previsión	Límite fijado y proyección del gasto. Avisar al alcanzar el límite llega tarde: el dinero ya se gastó.
desperdicio por abandono	Recursos que existen porque nadie los apagó. Es un problema de dueño, no de eficiencia.

Modelo mental

Gobernar no significa aprobar cada cambio: significa codificar límites, evidencia y responsabilidades para que los equipos puedan avanzar con seguridad.

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. Sin atribución no hay nada

La primera pregunta ante cualquier factura no es «cómo bajarla», sino de quién es.

```
si nadie sabe de quién es un gasto
  → nadie lo revisa
  → nadie lo puede reducir
```

```
   → y nadie lo echa de menos cuando ya no hace falta
```

Y las tres formas de atribuir, de más gruesa a más fina:

```
POR CUENTA O PROYECTO   la más fiable, porque no depende de nadie
                        un proyecto por equipo y entorno resuelve el 80 %
POR ETIQUETA            necesaria para el detalle
                        y depende de que se pongan
POR CONSUMO MEDIDO      para lo compartido: un clúster, una base multiuso
```

Y la segunda solo funciona si es obligatoria en la creación, con la política del proveedor de la clase 139: una etiqueta que se pide por convención no se pone.

```
etiquetas mínimas   equipo · servicio · entorno · centro de coste
y una más que ahorra discusiones:  fecha de caducidad prevista
```

Lo compartido, que es donde se atasca la conversación. Lo que no es de nadie en concreto:

```
red: pasarelas, tráfico entre zonas, salida         clase 135
nodos de un clúster con cargas de varios equipos
observabilidad                                      clase 121
plataforma y herramientas                           clase 106
soporte del proveedor y descuentos
```

Y los tres métodos de reparto:

```
A PARTES IGUALES        simple; injusto y nadie lo discute mucho
PROPORCIONAL AL USO     por peticiones, por CPU consumida, por GB
                        → el más defendible, y hay que poder medirlo
BOLSA SIN ASIGNAR       se deja aparte y lo asume la organización
                        → honesto, y si crece nadie lo reduce
```

Y la regla que evita meses de discusión:

```
el reparto tiene que ser EXPLICABLE, no perfectamente justo
→ un reparto que nadie entiende se discute y se ignora
→ y uno aproximado que se entiende cambia comportamientos
```

Y una advertencia sobre la bolsa sin asignar: conviene que sea pequeña y que tenga dueño. Si el 40 % de la factura no es de nadie, la mitad del trabajo no se puede hacer.

2. Mostrar, cobrar y el coste unitario

```
MOSTRAR    cada equipo ve su gasto y su evolución; no afecta a su presupuesto
  + suficiente en la mayoría de los casos
  + no genera discusiones sobre el reparto
  − si nadie mira, no cambia nada

COBRAR     el gasto se carga al presupuesto del equipo
  + cambia el comportamiento de verdad
  − obliga a que el reparto sea defendible
  − y produce optimizaciones locales que empeoran el total
```

Y la última línea es la ley 17 aplicada al dinero:

```
un equipo mueve una carga a otro sitio para que no le cuente
un equipo deja de usar la plataforma común porque le cobran su parte
y se monta la suya, que cuesta más en total
```

Y la recomendación práctica: mostrar casi siempre; cobrar cuando la organización ya sabe interpretar las cifras y con un reparto que los equipos acepten.

El coste unitario es lo que hace útil todo lo anterior:

```
coste absoluto      sube cuando el negocio va bien; no dice nada
coste por pedido    baja cuando se mejora y sube cuando se empeora
```

```
factura del mes                  41.000 €      →  52.000 €
pedidos del mes                 180.000        →   310.000
coste por pedido                  0,228 €      →     0,168 €
```

La factura sube un 27 % y la eficiencia mejora un 26 %. Sin la unidad, esa conversación no se puede tener.

Y las unidades que suelen servir:

```
por pedido, por transacción, por cliente activo
por petición servida, por gigabyte procesado
por sesión, por documento indexado
```

Y dónde se pone la cifra, que es lo que decide si sirve:

```
en el panel del servicio, junto al objetivo            clase 125
en el cambio propuesto: coste estimado del cambio
en la revisión de diseño: coste por unidad esperado
y en el catálogo del servicio                          clase 095
```

La segunda es la más eficaz: poner el coste delante en el momento de decidir cambia decisiones que ninguna revisión posterior cambiaría. Y no hace falta precisión: basta un orden de magnitud.

```
«este cambio añade una réplica de lectura: +310 €/mes»
«este índice nuevo: +0,004 € por pedido»
«guardar estos registros 90 días en vez de 14: +1.900 €/mes»
```

3. Avisar antes de gastarlo

El presupuesto con aviso al alcanzarlo llega tarde por definición: cuando avisa, el dinero está gastado. Lo que sirve:

```
aviso por PREVISIÓN     «al ritmo actual, se superará el presupuesto el día 21»
                        → avisa con margen para actuar
aviso por RITMO         gasto diario frente al esperado
                        → es el mismo mecanismo que el ritmo de consumo
                          del presupuesto de error de la clase 126
```

Y conviene tener presupuesto por equipo y por entorno, no solo global: un presupuesto global no lo mira nadie, porque no es de nadie.

La detección de desviaciones tiene un problema propio: el gasto es naturalmente irregular.

```
cierres mensuales, campañas, migraciones, procesos por lotes
→ un umbral simple produce falsas alarmas constantes
→ y entonces se desactiva: ley 16
```

Lo que funciona:

```
comparar cada servicio consigo mismo, no con el total
tener en cuenta el día de la semana y el del mes
avisar por CAMBIO relativo sostenido, no por valor absoluto
y agrupar por etiqueta, para que el aviso diga de quién es
```

Y el aviso útil dice tres cosas:

```
qué servicio, de qué equipo
cuánto más de lo esperado, en euros al día
y qué cambió: enlace a la línea de cambios de la clase 121
```

La tercera es la que convierte un aviso de coste en algo accionable en minutos en lugar de días.

Y la conexión con la ley 19, que es lo que hace peligroso el coste en la nube:

```
en un centro de datos propio, una fuga o un bucle acaban en una CAÍDA
  → se detecta en minutos
en la nube, el autoescalado lo absorbe
  → no hay caída: hay factura
  → y se detecta cuando alguien la mira, semanas después
```

Es exactamente el mecanismo que compensa un fallo y lo vuelve invisible, en versión financiera. Y de ahí dos consecuencias:

```
toda carga con autoescalado necesita TECHO          clases 117, 135
y el coste diario por servicio es una señal de
  funcionamiento, no solo de dinero
```

Y una cifra que conviene medir sobre el propio programa:

```
tiempo desde que empieza una desviación hasta que alguien la ve
  sin nada                semanas: cuando llega la factura
  con panel mensual       semanas
  con detección diaria    1-2 días
```

4. El desperdicio que es falta de dueño

Buena parte de lo que se llama «optimización» no es eficiencia: es retirar cosas que nadie apagó.

```
entornos de pruebas encendidos de noche y fin de semana
entornos efímeros huérfanos                         clase 104
discos sin conectar a nada
instantáneas antiguas                               clase 112
```

```
direcciones reservadas y no usadas
balanceadores sin nada detrás
bases de datos de proyectos terminados
cuentas enteras sin dueño                              clase 139
registros y métricas que nadie consulta                clase 121
licencias contratadas para un equipo que se disolvió
capacidad comprometida para un pico que ya no ocurre
```

Y todas tienen la misma causa: existen porque nadie es responsable de que dejen de existir. Por eso esta clase empieza por la atribución y no por el ahorro.

Los mecanismos que lo resuelven, todos ya vistos en el programa:

```
etiqueta de caducidad prevista, obligatoria al crear
barrido de huérfanos, programado                       clase 104
apagado automático de entornos inferiores fuera de horario
retirada por desuso, con recuperación inmediata        clase 134
y una regla que lo cierra: lo que no tiene dueño en 30 días se apaga
```

Y el orden de trabajo, que evita perder el tiempo:

```
1. lo que se puede apagar sin que nadie lo note        rendimiento inmediato
2. lo que se puede dimensionar mejor                   clase 143
3. lo que se puede comprometer con descuento           clase 143
4. lo que exige rediseño                               lo último
```

Y una advertencia sobre el punto 3, que es donde más se equivoca la gente: comprometerse a pagar durante un año algo que se podría apagar es peor que no hacer nada. Primero se retira y se dimensiona; solo después se compromete lo que quede.

Y lo que hay que vigilar de forma continua:

```
proporción de la factura sin atribuir
coste por unidad de negocio, por servicio
desviación diaria frente a lo previsto, por equipo
recursos sin dueño y su antigüedad
recursos con etiqueta de caducidad vencida
compromisos contratados frente a uso real
y la relación entre la factura y el coste de la telemetría
  que la mide                                          clase 132
```

Y la lista de comprobación de la clase:

- ■ las etiquetas de equipo, servicio y entorno son obligatorias al crear
- ■ hay un proyecto o cuenta por equipo y entorno
- ■ el reparto de lo compartido está escrito y es explicable
- ■ la bolsa sin asignar es pequeña y tiene dueño
- ■ cada equipo ve su gasto y su evolución
- ■ existe al menos una unidad de negocio y se publica el coste por unidad
- ■ el coste aparece en el cambio propuesto y en el panel del servicio
- ■ hay presupuesto por equipo, con aviso por previsión y por ritmo
- ■ la detección de desviaciones compara cada servicio consigo mismo
- ■ el aviso de desviación enlaza con la línea de cambios
- ■ toda carga con autoescalado tiene techo
- ■ hay barrido de huérfanos y apagado de entornos fuera de horario
- ■ se mide el tiempo desde que empieza una desviación hasta que se ve

Y el cierre que enlaza con la clase siguiente: con el gasto atribuido y visible, queda decidir qué hacer con él. Cómo se dimensiona, qué se compromete, qué se rediseña y qué relación tiene todo eso con el consumo energético es la materia de la clase 143.

Ejemplo trabajado

CloudShop tiene una factura de 41.000 € al mes que crece un 6 % mensual y nadie sabe explicar por qué. El ejercicio empieza por atribuir y termina encontrando que la mayor partida se compró para un pico que dejó de ocurrir hace catorce meses.

Punto de partida: la factura sin atribuir.

```
factura mensual                                      41.200 €
crecimiento mensual                                      6 %
proporción atribuible a un equipo                       31 %
proporción atribuible a un servicio                     22 %
```

```
cuentas y proyectos                                  23
recursos con etiqueta de equipo                   38 %
```

Sesenta y nueve por ciento del gasto no era de nadie.

La atribución, en tres semanas.

```
semana 1  política del proveedor: no se puede crear un recurso sin
          etiquetas de equipo, servicio, entorno y caducidad prevista
semana 2  campaña de etiquetado con el registro de auditoría
          (la misma de la clase 139, aprovechada)
semana 3  reparto de lo compartido, escrito y acordado
```

Y el reparto de lo compartido, que ocupó la mayor parte de la discusión:

```
nodos del clúster                  proporcional a CPU     medible
                                   solicitada
red: salida y entre zonas          proporcional a GB      medible
pasarelas de traducción            a partes iguales       no medible
observabilidad                     proporcional a series  medible
                                   y volumen
plataforma y herramientas          bolsa sin asignar      decisión
                                                          deliberada

gasto atribuible a un equipo                31 %            94 %
bolsa sin asignar                           69 %             6 %
```

El desglose, una vez atribuido.

```
capacidad comprometida sin usar                    9.800     24 %
observabilidad                                     6.410     16 %
cómputo de producción                              9.200     22 %
bases de datos                                     5.100     12 %
red                                                2.660      6 %
entornos inferiores                                3.900      9 %
almacenamiento y lago                              2.180      5 %
resto                                              1.950      5 %
```

La primera línea no la esperaba nadie.

La partida mayor: un pico que ya no ocurre.

```
compromiso contratado hace 26 meses, a 3 años
motivo original    una campaña anual que exigía 3× la capacidad
uso real del compromiso                                  41 %
el pico que lo motivó                  dejó de ocurrir hace 14 meses
                                       (cambió el calendario comercial)
quién lo revisaba                      nadie desde la firma
coste del compromiso no usado                      9.800 €/mes
```

Veinticuatro por ciento de la factura pagando capacidad para un evento que ya no existe. Y no era un error técnico: era un contrato que nadie tenía asignado.

```
compromiso                         3 años, 26 meses    renegociado al
                                   de antigüedad       uso real
uso del compromiso                       41 %            89 %
coste mensual del compromiso           9.800 €         4.100 €
ahorro                                    —             5.700 €/mes
revisión del compromiso                ninguna         trimestral,
                                                       con dueño
```

Los tres siguientes ahorros, para comparar.

```
telemetría (clase 132)                              5.300 €/mes
entornos inferiores apagados fuera de horario       1.900 €/mes
recursos huérfanos retirados                        1.140 €/mes
                                                    ■■■■■■■■■■■■
suma de los tres siguientes                         8.340 €/mes
```

Y el dato que la clase 144 usará para calificar la hipótesis de la clase 132:

```
mayor ahorro individual (compromiso)          5.700 €/mes
suma de los tres siguientes                   8.340 €/mes
→ el mayor NO supera a la suma de los tres siguientes
```

Los huérfanos, que son atribución y no eficiencia.

```
al barrer con las etiquetas ya puestas
```

```
  discos sin conectar                                  118
  instantáneas de más de 1 año                         940
  direcciones reservadas sin uso                        41
  balanceadores sin destinos                            11
  bases de datos de proyectos terminados                 4
  entornos efímeros huérfanos                  ya resuelto (clase 104)
  cuentas sin dueño                                      6  (clase 139)
                                                ■■■■■■■■■
  ahorro                                         1.140 €/mes
```

Y la regla que lo mantiene: lo que no tiene dueño en 30 días se apaga, con aviso previo y recuperación en minutos. En seis meses se apagaron 31 recursos y hubo 3 reclamaciones, todas resueltas el mismo día.

El coste unitario, y la conversación que permitió.

```
factura mensual                      41.200 €   28.900 €
pedidos                              180.000    310.000
coste por pedido                      0,229 €    0,093 €
```

Y la cifra por servicio reveló algo que la factura global escondía:

```
servicio de pedidos                          0,021 €
catálogo                                     0,014 €
recomendaciones                              0,038 €   ← el más caro
búsqueda                                     0,009 €
```

Recomendaciones costaba más por pedido que el propio servicio de pedidos. Nadie lo sabía, y llevó a un rediseño que lo bajó a 0,011 €.

La desviación que el autoescalado tapó.

```
día 4    un cambio introduce una consulta sin índice en el catálogo
         la latencia sube ligeramente; el autoescalado añade instancias
         no hay caída, ni alerta, ni queja
día 5-17 el servicio funciona con 3,4× las instancias habituales
día 18   la detección diaria de desviación avisa:
         «catálogo: +190 €/día sobre lo previsto desde el día 4»
día 18   la línea de cambios señala el despliegue del día 4
día 18   corregido en 40 minutos
coste de los 14 días                                  2.660 €
```

Es la ley 19 en versión financiera: en un sistema sin elasticidad habría sido una caída de minutos; con elasticidad fue una factura de dos semanas.

```
detección de desviaciones                    mensual        diaria
comparación                                con el total    cada servicio
                                                           consigo mismo
falsas alarmas por estacionalidad          no aplicable    2 en 6 meses
tiempo desde que empieza una desviación
hasta que se ve                            3-5 semanas      1,2 días
aviso enlazado con la línea de cambios        no             sí
techos de autoescalado definidos            0 de 15        15 de 15
```

Mostrar en vez de cobrar.

```
decisión    mostrar el gasto a cada equipo, sin cargarlo a su presupuesto
motivo      el reparto de lo compartido llevaba tres semanas de acuerdo
            y cobrar habría reabierto la discusión

efecto medido en 6 meses
  equipos que redujeron su gasto sin que nadie se lo pidiera      4 de 6
  reducción atribuible a esas decisiones                    3.100 €/mes
  discusiones sobre el método de reparto                          2
```

Y el coste en el cambio propuesto, que resultó más eficaz que el panel:

```
cambios en los que apareció una estimación de coste             211
cambios modificados tras verla                                   19
ahorro estimado de esas 19 decisiones                     1.400 €/mes
```

A los seis meses.

```
factura mensual                          41.200 €       28.900 €
coste por pedido                          0,229 €        0,093 €
gasto atribuible a un equipo                31 %           94 %
bolsa sin asignar                           69 %            6 %
uso del compromiso contratado               41 %           89 %
```

```
recursos sin dueño                          no medido          0
tiempo hasta ver una desviación           3-5 semanas       1,2 días
techos de autoescalado                      0 de 15         15 de 15
coste visible en el cambio propuesto           no              sí
revisión de compromisos                     ninguna         trimestral
```

La lección que esta clase traslada a la parte 11: la mayor partida de la factura —el 24 %— era capacidad comprometida a tres años para una campaña que dejó de celebrarse hace catorce meses, y llevaba dos años sin que nadie la revisara porque el contrato no tenía dueño asignado. No fue un problema de ingeniería ni de precios: fue exactamente el mismo problema que la clase 139 encontró con los recursos y la 141 con los sistemas que guardan datos personales. Nadie sabía de quién era cada cosa.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-11-security-governance-finops/142-finops-showback-chargeback-budgets-y-anomalias/lab.py
```

El laboratorio selecciona el motor de práctica finops 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 modelo-asignacion-costos 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 cálculo trazable con unidad, supuesto y sensibilidad. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye modelo-asignacion-costos 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 sabe cuánto se gasta y no de quién es	Las etiquetas se piden por convención y no se ponen	Impónlas con política del proveedor en la creación, y usa un proyecto o cuenta por equipo y entorno como atribución gruesa.
El reparto de los costes compartidos se discute durante meses	Se busca un reparto justo en vez de uno explicable	Elige un método defendible, escríbelo, y deja en una bolsa pequeña y con dueño lo que no se pueda repartir.
La factura sube y no se sabe si el sistema es más o menos eficiente	Solo se mira el coste absoluto, que crece con el negocio	Publica el coste por unidad de negocio, global y por servicio.
El aviso de presupuesto llega cuando ya se gastó	Se avisa al alcanzar el límite	Avisa por previsión y por ritmo de gasto, con presupuesto por equipo y no solo global.
Una avería no provoca ninguna caída y aparece semanas después en la factura	Ley 19: el autoescalado la compensa y la vuelve invisible	Techo en toda carga elástica, detección diaria de desviación por servicio y aviso enlazado con la línea de cambios.
Se contrata un compromiso con descuento y el ahorro no llega	Se comprometió capacidad que se podía apagar o dimensionar mejor	Retira y dimensiona primero; compromete solo lo que quede, y revisa el compromiso cada trimestre con un dueño asignado.

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 la atribución es el requisito previo de todo lo demás?
2. ¿Qué regla evita meses de discusión sobre el reparto de lo compartido?
3. ¿Por qué el coste por unidad de negocio dice cosas que el coste absoluto no dice?
4. ¿Por qué avisar al alcanzar el presupuesto llega tarde y qué se hace en su lugar?
5. ¿Cómo convierte la elasticidad una avería en una factura, y qué lo detecta?

Referencias

- FinOps Foundation (2025). FinOps framework: capabilities — atribución, visibilidad, optimización y gobierno. <https://www.finops.org/framework/>
- FinOps Foundation (2025). Unit economics — coste por unidad de negocio como medida comparable. <https://www.finops.org/framework/capabilities/unit-economics/>
- AWS (2025). Cost allocation tags and cost anomaly detection — atribución y detección de desviaciones. <https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/cost-alloc-tags.html>
- Google Cloud (2025). Billing budgets, forecasts and alerts — aviso por previsión frente a aviso por límite. <https://cloud.google.com/billing/docs/how-to/budgets>
- Azure (2025). Cost Management: showback and chargeback — mostrar frente a cobrar y sus efectos. <https://learn.microsoft.com/azure/cost-management-billing/finops/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 11 en PDF · Manual integral

Evaluación — 142 FinOps: showback, chargeback, budgets y anomalías

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 modelo-asignacion-costos 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.

143 — Optimización de costo, capacidad y sostenibilidad

Parte: 11 — Seguridad, gobierno, cumplimiento y FinOps Nivel: avanzado · Horas estimadas: 4 Laboratorio: finops · Estado: EXECUTABLECORE

Propósito

Decidir qué hacer con el gasto una vez atribuido, en un orden que evita el error más caro de esta materia: comprometerse a pagar durante tres años algo que se podía haber apagado. La clase desarrolla la escalera —eliminar, dimensionar, programar, comprometer, rediseñar—, trata el dimensionado con la honestidad que exige la curva de la clase 129, y cierra con la parte energética sin exagerarla: casi toda la optimización de coste es también de consumo, y conviene saber en qué casos concretos no coinciden.

Resultados de aprendizaje

Al finalizar podrás:

1. Aplicar la escalera de optimización en orden y no saltarse escalones.
2. Dimensionar con percentiles y margen, no con medias.
3. Combinar instrumentos de descuento por capas según la estabilidad del consumo.
4. Usar capacidad interrumpible en lo que la tolera, y solo ahí.
5. Separar lo que reduce consumo energético de lo que solo reduce factura.

Conceptos centrales

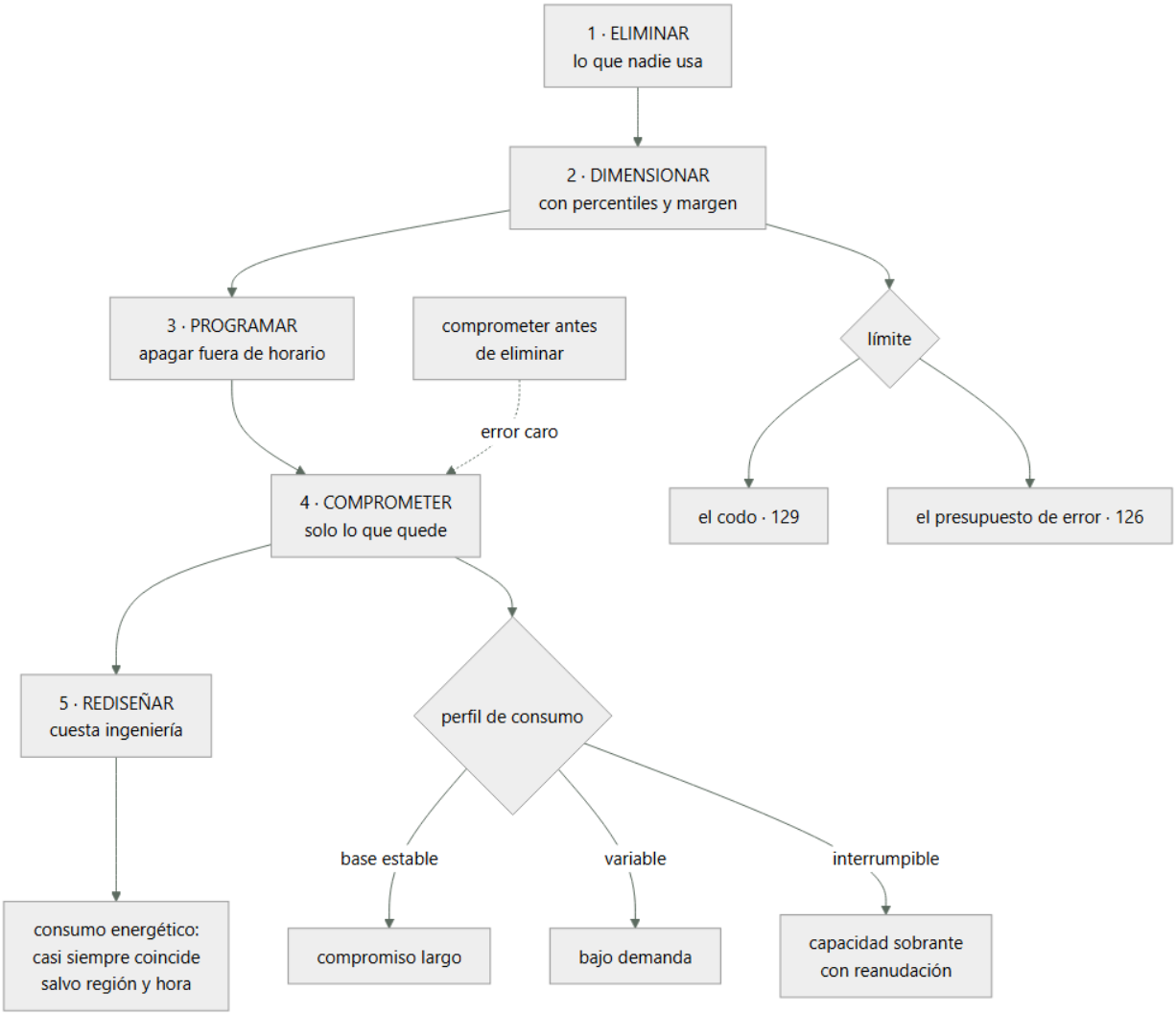
Concepto	Comprensión verificable
escalera de optimización	Orden en el que se trabaja: eliminar, dimensionar, programar, comprometer y rediseñar. Saltárselo cuesta dinero.
dimensionado	Ajustar recursos al uso real, con margen hasta el codo. Se hace con percentiles, nunca con medias.
compromiso	Descuento a cambio de garantizar un gasto durante un periodo. Es una apuesta sobre tu propia arquitectura.
capacidad interrumpible	Recursos muy baratos que el proveedor puede retirar con poco aviso. Sirven solo para cargas que toleran la interrupción.
intensidad de carbono	Emisiones por unidad de energía consumida. Varía por región y por hora, y es donde coste y emisiones dejan de coincidir.
tensión eficiencia-fiabilidad	Reducir margen abarata y acerca al codo. El presupuesto de error decide el límite.

Modelo mental

Gobernar no significa aprobar cada cambio: significa codificar límites, evidencia y responsabilidades para que los equipos puedan avanzar con seguridad.

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 escalera, y por qué el orden importa

```
1. ELIMINAR      lo que nadie usa              clase 142
2. DIMENSIONAR   ajustar lo que sí se usa
3. PROGRAMAR     apagar cuando no hace falta
4. COMPROMETER   descuento sobre lo que queda
5. REDISEÑAR     cambiar cómo funciona
```

Y el orden no es una preferencia: cada escalón cambia la base sobre la que se calcula el siguiente.

```
comprometer antes de eliminar
  → se firma un descuento sobre capacidad que se iba a apagar
  → y ese compromiso impide apagarla, porque ya está pagada
  → resultado: el descuento cuesta dinero
```

Es el error de la clase 142 en su forma general, y ocurre porque los escalones 1 a 3 requieren trabajo y el 4 se firma en una tarde.

Y el rendimiento típico de cada escalón, como orden de magnitud:

```
eliminar        10-30 % de la factura, sin riesgo y sin ingeniería
dimensionar     10-25 %, con riesgo si se hace mal
programar       5-15 %, concentrado en entornos inferiores
comprometer     20-40 % sobre lo que quede, con compromiso de gasto
rediseñar       muy variable, y cuesta semanas de ingeniería
```

Y una conclusión que sorprende: los tres primeros escalones suelen dar más que el descuento, y no requieren negociar nada.

Y el quinto merece un criterio explícito, porque es donde se pierde el tiempo:

```
rediseñar compensa cuando el coste CRECE con el tamaño del negocio
no compensa cuando es un coste fijo pequeño
→ optimizar una consulta que cuesta 90 € al mes no vale dos semanas
→ optimizar el coste por pedido, sí
```

Y el límite que atraviesa toda la escalera, y que hay que enunciar antes de empezar:

```
ninguna medida de eficiencia se aplica si consume el presupuesto
de error                                              clase 126
→ el margen no es desperdicio: es el tiempo de reaccionar
```

2. Dimensionar sin romper

El dimensionado es el escalón con más riesgo, porque acerca el sistema al codo de la clase 129.

Y el error habitual es usar medias:

```
uso medio de procesador                    12 %
conclusión ingenua                         «sobra el 80 %»
uso en el percentil 99                     71 %
uso durante el cierre mensual              94 %
```

La media de un sistema con picos no describe nada. La regla:

```
se dimensiona con el percentil alto del periodo relevante
incluyendo los eventos periódicos: cierres, campañas, procesos nocturnos
y dejando el margen hasta el codo                      clase 129
```

Y qué mirar, por recurso:

```
procesador        percentil 95-99, y el pico de los eventos conocidos
memoria           el máximo, no el percentil: quedarse sin memoria
                  no degrada, mata el proceso
disco             espacio y también operaciones por segundo
red               ancho de banda del pico
conexiones        saturación del agrupador                clase 109
```

La segunda línea es una diferencia importante: la memoria no admite percentiles.

Y dos consideraciones específicas de la nube:

```
MEMORIA Y PROCESADOR VAN ACOPLADOS en muchos servicios
  → bajar memoria puede bajar el procesador y empeorar el tiempo
  → hay que medir el resultado, no suponerlo         clase 117

UNA FAMILIA DISTINTA puede ser mejor que un tamaño menor
  → cargas ligadas a memoria, a procesador o a red tienen familias propias
  → cambiar de familia suele rendir más que reducir el tamaño
```

Programar es el escalón más sencillo y el que menos se hace:

```
entornos inferiores fuera de horario laboral
  → 168 horas a la semana → 45 útiles → 73 % de ahorro en esos entornos
bases de datos de desarrollo, paradas por la noche
capacidad reducida en horas valle en producción, si el perfil lo permite
```

Y lo que lo hace viable: encender tiene que ser rápido y automático, o alguien lo desactivará el primer día que necesite trabajar por la tarde. Es la ley 16 aplicada al ahorro.

3. Comprometer por capas, y lo interrumpible

Los instrumentos de descuento cambian de nombre en cada proveedor y se comportan igual:

```
COMPROMISO DE GASTO O DE CAPACIDAD
  1 o 3 años, con o sin pago anticipado
  descuento mayor cuanto más rígido y más largo
  → es una apuesta a que tu arquitectura seguirá pareciéndose

CAPACIDAD SOBRANTE INTERRUMPIBLE
  60-90 % más barata
  el proveedor la retira con un aviso de minutos

NIVELES Y TARIFAS POR VOLUMEN
  descuentos automáticos por uso acumulado
```

Y la forma correcta de usarlos es por capas, según la estabilidad de cada tramo:

```
consumo del último año, por horas, ordenado de mayor a menor

  ■■■■■■■■■■■■■■■■■■■■■■■■  base que SIEMPRE está    → compromiso a 3 años
  ■■■■■■■■■■■                lo habitual pero variable → compromiso a 1 año
  ■■■■                      picos                      → bajo demanda
  ■■                        trabajos que toleran corte → interrumpible
```

Y las reglas prácticas:

```
no comprometer más del percentil bajo del consumo estable
  → comprometer al 100 % del uso actual es garantizar pagar de más
revisar cada trimestre, con dueño                        clase 142
y anotar la fecha de vencimiento como un riesgo, no como un trámite
```

Y el riesgo que hay que decir en voz alta: un compromiso a tres años es una apuesta sobre tu propia arquitectura. Si en dieciocho meses se migra a contenedores, a funciones o a otro proveedor, el compromiso sigue ahí.

La capacidad interrumpible es el descuento mayor y exige que la carga lo tolere:

```
requisitos
  soportar que la instancia desaparezca con un aviso de minutos
  poder reanudar: puntos de control, o trabajo troceado e idempotente
  diversificar tipos y zonas, para que no se retire todo a la vez

encaja bien
  construcción y pruebas de la canalización              clase 097
  procesamiento por lotes y trabajos del lago            clase 112
  entornos efímeros                                      clase 104
  cargas sin estado con arranque rápido y un grupo de respaldo

no encaja
  bases de datos y cualquier cosa con estado local
  nada cuyo corte se note en el objetivo del servicio    clase 126
```

Y el error clásico: usar capacidad interrumpible para algo que no puede reanudarse, y descubrirlo el día que el proveedor retira el 40 % de golpe. La prueba pertinente es la de la clase 131: quitar instancias a propósito y ver qué pasa.

4. Rediseñar, datos y consumo energético

Los rediseños con mayor multiplicador, todos ya vistos en el programa:

```
caché delante de lo caro                                 clase 111
quitar llamadas repetidas: una consulta en vez de 340    clase 124
agrupar y procesar por lotes en vez de uno a uno         clase 112
llevar el cálculo al dato, y no el dato al cálculo
formato columnar y particionado                          clase 112
nivel de servicio adecuado: no todo necesita el más caro
y quitar funcionalidad que nadie usa
```

El último es el de mayor rendimiento y el que nunca se propone: una funcionalidad sin usuarios cuesta cómputo, almacenamiento, mantenimiento y superficie de ataque.

En datos, donde suele haber más grasa de la que parece:

```
retención: ¿de verdad hacen falta 90 días calientes?     clases 122, 123
clases de almacenamiento, agrupando antes                clase 112
compresión y formato                                     clase 112
borrar lo que nadie consulta
copias de seguridad: cuántas generaciones y de qué
y entornos de pruebas con subconjuntos, no con copias completas
```

El consumo energético. Conviene tratarlo sin exagerar y sin ignorarlo:

```
la energía consumida es, en primera aproximación,
proporcional a los recursos-hora usados
→ por tanto, casi toda optimización de coste reduce también consumo
→ eliminar, dimensionar y programar valen doble
```

Y los casos donde no coinciden, que son los específicos de esta materia:

```
REGIÓN
  la intensidad de carbono varía mucho entre regiones
  → dos regiones con precio parecido pueden diferir por un factor grande
```

```
  → y la región también la deciden la latencia y la residencia (clase 141)

HORA
  la intensidad varía a lo largo del día según la generación
  → mover trabajos por lotes a horas de baja intensidad no cambia
    el coste y sí las emisiones

EFICIENCIA DEL HARDWARE
  familias más modernas hacen más trabajo por vatio
  → suele coincidir con mejor precio por unidad de trabajo

APROVECHAMIENTO
  una instancia al 15 % consume bastante más de un 15 %
  → consolidar cargas reduce energía más de lo que reduce coste
```

Y la parte honesta sobre la medición:

```
las cifras las publica el proveedor, con metodologías distintas
no son comparables entre proveedores
y la mayor parte de las emisiones de un servicio digital suele
  estar fuera del cómputo: fabricación, red, dispositivos de usuario
→ conviene usarlas para decidir entre alternativas propias,
  no para afirmaciones absolutas
```

Y la tensión final, que cierra la clase:

```
reducir margen abarata y acerca al codo
reducir réplicas abarata y reduce tolerancia a fallos
reducir retención abarata y reduce capacidad de investigar

→ el límite lo pone el presupuesto de error, no el objetivo de ahorro
→ si una medida de eficiencia consume presupuesto, se revierte
```

Y la lista de comprobación de la clase:

- [ ] se ha eliminado lo no usado antes de dimensionar
- [ ] se ha dimensionado antes de comprometer
- [ ] el dimensionado usa percentiles altos y el máximo para memoria
- [ ] se han probado varias familias, no solo tamaños menores
- [ ] los entornos inferiores se apagan fuera de horario y encender es rápido
- [ ] los compromisos cubren la base estable, no el consumo total
- [ ] los compromisos se revisan cada trimestre y tienen dueño
- [ ] la capacidad interrumpible solo se usa donde la interrupción se tolera
- [ ] se ha ensayado la retirada de capacidad interrumpible
- [ ] los rediseños se eligen por si el coste crece con el negocio
- [ ] está medido el coste por unidad antes y después de cada medida
- [ ] ninguna medida de eficiencia consume presupuesto de error
- [ ] las decisiones de región consideran también intensidad de carbono

Y el cierre que enlaza con la clase siguiente: con esto está completo el material de la parte 11. La clase 144 monta la zona de aterrizaje con todo lo anterior aplicado desde el primer día y califica las cinco predicciones de la clase 132, empezando por la que decía que la ley 16 dominaría esta parte.

Ejemplo trabajado

CloudShop aplica la escalera sobre los 28.900 € que quedaban tras la clase 142. El ejercicio incluye un dimensionado que provocó un incidente y una decisión de compromiso que estuvo a punto de repetir el error de partida.

Escalón 1: eliminar. Ya hecho en la clase 142.

```
compromiso mal dimensionado                    −5.700 €/mes
telemetría sin consultar                       −5.300 €/mes
huérfanos                                      −1.140 €/mes
```

Escalón 2: dimensionar, y el incidente.

El primer intento usó medias:

```
15 servicios, uso medio de procesador                    14 %
propuesta automática de la herramienta     reducir a la mitad en 11
aplicado                                                  11
```

Y a los cuatro días:

```
cierre mensual: el proceso de facturación necesita 3× la capacidad
```

```
servicios afectados                                  3
latencia p99                                 de 180 ms a 4,1 s
presupuesto de error consumido                      31 %
revertido en                                      22 min
```

El diagnóstico es el del apartado segundo: la media no incluía el evento periódico.

```
servicios reducidos                    11                  7
reducción media aplicada               50 %               28 %
ahorro                              1.900 €/mes        1.180 €/mes
incidentes                             1                   0
presupuesto de error consumido         31 %                0 %
```

Setecientos euros menos de ahorro y ningún incidente. Y se añadió una regla: ninguna propuesta de dimensionado se aplica sin incluir los eventos periódicos conocidos, que están en el calendario del negocio.

Y el cambio de familia, que rindió más que reducir tamaño:

```
                                                    para memoria
servicio de catálogo, coste          −22 %              −34 %
latencia p99                         +41 %               −8 %
```

Escalón 3: programar.

```
entornos inferiores                        3 (dev, pre, pruebas)
horas encendidos antes                   168 / semana
horas encendidos después                  50 / semana
ahorro                                 2.100 €/mes

condición para que se aceptara
  encender bajo demanda en menos de 3 minutos
  y encendido automático al abrir un cambio propuesto
```

Y sin esa condición no habría durado: en la primera semana, dos equipos desactivaron el apagado porque necesitaban trabajar por la tarde. Con el encendido en tres minutos, nadie volvió a desactivarlo.

Escalón 4: comprometer, y la trampa evitada.

La propuesta inicial del proveedor cubría el consumo completo del último mes:

```
propuesta              compromiso a 3 años sobre el 100 % del uso
descuento ofrecido                                  41 %
ahorro aparente                                6.100 €/mes
```

Y la revisión con las reglas del apartado tercero encontró dos problemas:

```
1. el uso del último mes incluía lo que se acababa de dimensionar
   y los entornos que ahora se apagan
   → el consumo real ya era un 31 % menor

2. había una migración prevista a contenedores en 8 meses
   para 4 de los 15 servicios
   → ese consumo cambiaría de forma
```

```
base comprometida a 3 años          100 % del uso      62 % (la base
                                    del mes anterior    estable medida
                                                        en 12 meses)
compromiso a 1 año                        —              18 %
bajo demanda                              0 %            12 %
interrumpible                             0 %             8 %
descuento efectivo                       41 %            34 %
ahorro real                         6.100 € aparente   4.200 €/mes
riesgo de pagar capacidad no usada         alto            bajo
```

Menos descuento nominal y más ahorro real, porque no se compromete lo que se va a apagar.

La capacidad interrumpible.

```
cargas evaluadas                                       15
toleran interrupción                                    4
  construcción y pruebas de la canalización            sí
  trabajos del lago                                    sí
  entornos efímeros                                    sí
  reprocesado de eventos                               sí

ahorro en esas cuatro                            1.400 €/mes
```

Y el ensayo de la clase 131, hecho antes de confiar en ello:

```
se retiró el 60 % de la capacidad interrumpible a propósito

construcción y pruebas    reanudaron; +9 min en la canalización     ✓
trabajos del lago         reanudaron desde el último punto           ✓
entornos efímeros         se recrearon solos                         ✓
reprocesado de eventos    PERDIÓ el progreso: no había punto
                          de control; 4 h de trabajo repetido        ✗
```

El cuarto se corrigió antes de dejarlo en interrumpible. Sin el ensayo, se habría descubierto un domingo.

Escalón 5: rediseñar, con el criterio del coste unitario.

```
candidatos                                 coste actual   ¿crece con el negocio?
recomendaciones, 0,038 €/pedido              4.100 €/mes         sí   → hacer
consulta del catálogo                           90 €/mes         no   → no hacer
formato del lago (ya columnar)                   —               —   → hecho
llamadas repetidas en 2 endpoints             310 €/mes          sí   → hacer

coste por pedido de recomendaciones            0,038 €        0,011 €
esfuerzo                                          —          3 semanas
ahorro                                            —        2.900 €/mes
```

Y la consulta de 90 € al mes no se optimizó, deliberadamente y por escrito.

El consumo energético.

```
recursos-hora consumidos                    −38 % tras la escalera
emisiones estimadas por el proveedor        −41 %
```

Y las dos decisiones específicas, que no cambiaron el coste:

```
REGIÓN de los trabajos por lotes
  se movieron los del lago a una región con menor intensidad de carbono
  restricción: los datos personales no salen de Europa   clase 141
  → solo pudieron moverse los agregados sin datos personales
  coste                                                  +2 %
  emisiones estimadas de esos trabajos                  −34 %

HORA de ejecución
  los procesos nocturnos se desplazaron 3 horas
  coste                                                  igual
  emisiones estimadas                                   −11 %
```

Y lo que se escribió como límite honesto:

```
las cifras las publica el proveedor, con su metodología
no son comparables con las de otro proveedor
y la mayor parte de las emisiones del producto está fuera del cómputo
→ se usan para elegir entre alternativas propias, no para afirmar nada
  en términos absolutos
```

El resultado de la escalera.

```
punto de partida (tras la clase 142)              28.900
dimensionar                                       −1.180
programar                                         −2.100
comprometer                                       −4.200
rediseñar                                         −2.900
                                                  ■■■■■■■
                                                  18.520

factura mensual                              41.200 €       18.520 €
coste por pedido                              0,229 €        0,057 €
pedidos mensuales                             180.000        325.000
recursos-hora                                    —            −38 %
incidentes causados por optimización             —              1
presupuesto de error consumido por eficiencia    —          31 % (una vez)
compromisos revisados trimestralmente           no             sí
cargas en capacidad interrumpible                0              4
ensayo de retirada de interrumpible             no         semestral
```

La lección que esta clase traslada a la parte 11: la propuesta de compromiso que ofrecía un 41 % de descuento habría producido menos ahorro real que la que ofrecía un 34 %, porque cubría capacidad que estaba a punto de desaparecer. Y el único incidente de todo el ejercicio lo causó dimensionar con medias en un sistema con cierres mensuales: la

media de un sistema con picos no describe nada, y aplicarla consumió el 31 % del presupuesto de error para ahorrar setecientos euros.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-11-security-governance-finops/143-optimizacion-de-costo-capacidad-y-sostenibilidad/lab.py
```

El laboratorio selecciona el motor de práctica finops 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 backlog-optimizacion 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 cálculo trazable con unidad, supuesto y sensibilidad. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye backlog-optimizacion 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 firma un descuento y el ahorro esperado no aparece	Se comprometió capacidad que se iba a eliminar, dimensionar o apagar	Recorre la escalera en orden: elimina, dimensiona y programa antes de comprometer, y compromete solo la base estable medida en doce meses.
Reducir tamaños provoca un incidente en el cierre mensual	Se dimensionó con la media, que no incluye los eventos periódicos	Usa percentiles altos, incluye los eventos del calendario de negocio y el máximo para memoria.
El apagado de entornos fuera de horario se desactiva a la semana	Ley 16: encender es lento y estorba	Haz que encender tarde minutos y sea automático al abrir un cambio propuesto.
El proveedor retira capacidad interrumpible y se pierde trabajo	Se usó para una carga que no puede reanudarse	Exige puntos de control o trabajo troceado e idempotente, diversifica tipos y zonas, y ensaya la retirada antes de confiar.
Se dedican semanas a optimizar algo que apenas cuesta	No se distingue el coste fijo del que crece con el negocio	Rediseña solo lo que aparece en el coste por unidad; deja documentado lo que se decide no optimizar.
La eficiencia mejora y la fiabilidad empeora	Se recortó el margen hasta el codo	El presupuesto de error es el límite: si una medida de eficiencia lo consume, se revierte.

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 comprometer antes de eliminar cuesta dinero?
2. ¿Por qué la media no sirve para dimensionar y qué se usa en su lugar?
3. ¿Cómo se reparte el consumo entre compromiso, bajo demanda e interrumpible?
4. ¿Qué requisitos tiene una carga para poder usar capacidad interrumpible?
5. ¿En qué casos concretos dejan de coincidir la reducción de coste y la de emisiones?

Referencias

- FinOps Foundation (2025). Rate and usage optimization — escalera de optimización y orden de aplicación. <https://www.finops.org/framework/capabilities/>
- AWS (2025). Savings Plans, Reserved Instances and Spot best practices — instrumentos y su combinación por capas. <https://docs.aws.amazon.com/whitepapers/latest/cost-optimization-reservation-models/>
- Google Cloud (2025). Committed use discounts and Spot VMs — compromisos y capacidad interrumpible. <https://cloud.google.com/compute/docs/instances/spot>
- Green Software Foundation (2025). Software carbon intensity specification — intensidad por región y por hora. <https://sci.greensoftware.foundation/>
- Microsoft (2025). Sustainability in the cloud: emissions reporting caveats — límites de comparabilidad de las cifras. <https://learn.microsoft.com/azure/architecture/framework/sustainability/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 11 en PDF · Manual integral

Evaluación — 143 Optimización de costo, capacidad y sostenibilidad

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 backlog-optimizacion 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.

144 — Proyecto: landing zone con guardrails

Parte: 11 — Seguridad, gobierno, cumplimiento y FinOps Nivel: avanzado · Horas estimadas: 8 Laboratorio: capstone · Estado: EXECUTABLECORE

Propósito

Montar la base sobre la que se despliega todo lo demás —estructura de cuentas, identidad, red, controles preventivos, registro, atribución y presupuestos— con lo de las clases 133 a 143 aplicado desde el primer día en vez de añadido después. Y cerrar la parte con las tres piezas de siempre: calificar las cinco predicciones de la clase 132, una de las cuales falló y otra se cumplió de forma más amplia de lo previsto; incorporar la ley que esta parte ha hecho aparecer cinco veces; y escribir la predicción que la parte 12 tendrá que corregir.

Resultados de aprendizaje

Al finalizar podrás:

1. Ordenar los elementos de una base de nube por lo que hacen imposible después.
2. Montar la estructura con controles preventivos y atribución obligatoria.
3. Calificar las cinco predicciones de la clase 132 con evidencia.
4. Incorporar la ley 20 al cuestionario, con sus cinco apariciones.
5. Escribir la predicción de la parte 12 en términos que se puedan desmentir.

Conceptos centrales

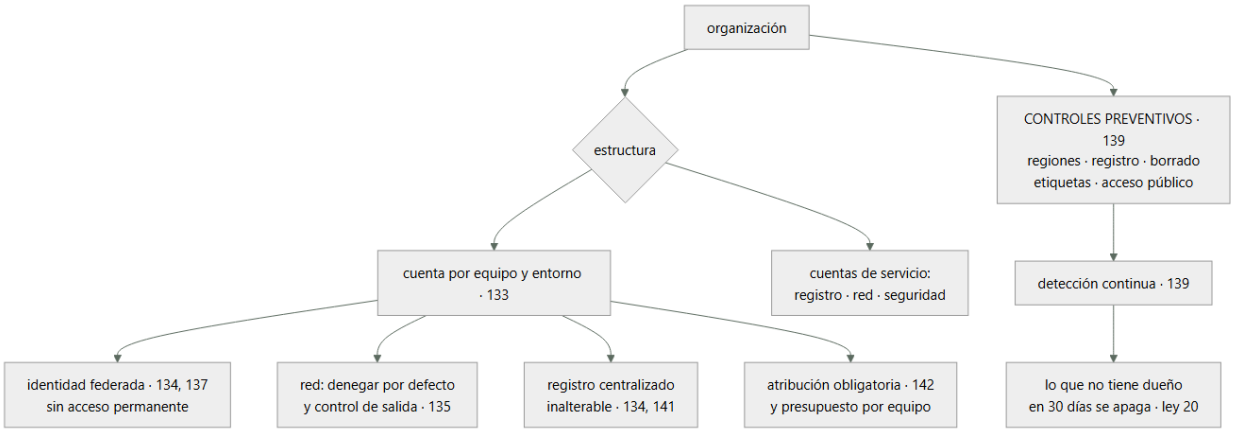
Concepto	Comprensión verificable
zona de aterrizaje	Base preconfigurada donde se despliegan las cargas: estructura, identidad, red, controles, registro y atribución.
control preventivo	Política que impide una acción, aunque quien la intente tenga permisos. Es lo único que no se rodea.
atribución obligatoria	Etiquetas de dueño impuestas en la creación. Sin ellas, la seguridad y el coste comparten el mismo agujero.
ley 20	La falta de dueño es la causa común de la fuga y del despilfarro: lo que no tiene responsable no se apaga, no se corrige y no se retira.
calificación de hipótesis	Comparar lo predicho con lo ocurrido, publicando también lo que se predijo mal.
hipótesis de la parte 12	Predicción escrita ahora sobre lo que ocurrirá cuando el sujeto sea la arquitectura y sus decisiones.

Modelo mental

Gobernar no significa aprobar cada cambio: significa codificar límites, evidencia y responsabilidades para que los equipos puedan avanzar con seguridad.

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 va en la base y en qué orden

Una zona de aterrizaje no es un producto: es el conjunto de decisiones que después no se pueden cambiar sin migrar. Por eso el criterio de orden es cuál hace imposible a las demás.

```
1. ESTRUCTURA DE CUENTAS O PROYECTOS
   una por equipo y entorno, más las de servicios comunes
   → es la frontera más fuerte que existe            clase 133
   → y cambiarla después es mover todo lo que contiene

2. CONTROLES PREVENTIVOS DE LA ORGANIZACIÓN
   lo que nunca debe ocurrir, aunque haya permisos    clase 139
   → regiones permitidas, registro que no se puede desactivar,
     borrado de copias prohibido, etiquetas obligatorias,
     acceso público bloqueado

3. IDENTIDAD
   federación para personas y para cargas             clases 134, 137
   sin acceso permanente a lo sensible
   sin claves de larga duración

4. RED
   denegar por defecto entre servicios                clase 135
   puntos de acceso privados y control de salida

5. REGISTRO Y EVIDENCIA
   auditoría centralizada, en una cuenta que nadie más puede tocar
   inalterable                                        clases 134, 141

6. ATRIBUCIÓN Y PRESUPUESTO
   etiquetas impuestas al crear, presupuesto por equipo
                                                      clase 142

7. DETECCIÓN CONTINUA
   sobre todo lo anterior, incluida la comprobación de que sigue activo
                                                      clase 139
```

Y las tres decisiones de esa lista que son irreversibles en la práctica:

```
la estructura de cuentas
el dominio de identidad y su federación
y el esquema de etiquetado
```

Y una regla que ahorra rehacerlo todo al segundo año: la base se declara como infraestructura desde el primer día. Una zona de aterrizaje montada a mano no se puede reproducir, ni auditar, ni corregir en el origen.

Y el error más común al montarla, que conviene enunciar:

```
hacerla tan restrictiva que los equipos pidan excepciones para todo
→ y entonces la excepción se convierte en el camino normal
→ ley 16, otra vez
```

Y la contramedida es la de la clase 106: la base tiene que ser el camino más rápido, con un procedimiento de creación de proyecto que tarde minutos y traiga todo montado.

2. El proyecto

Montar la base y comprobarla. Lo que hay que entregar:

```
1. ESTRUCTURA declarada, con jerarquía y nomenclatura
   y un procedimiento de alta de proyecto que tarde minutos

2. CONTROLES PREVENTIVOS, con prueba negativa cada uno
   regiones · registro · borrado de copias · acceso público
   etiquetas obligatorias · claves de larga duración prohibidas

3. IDENTIDAD
   federación para personas y cargas
   concesión temporal más rápida que cualquier atajo
   acceso de emergencia preparado, con revisión obligatoria
   fronteras de permisos

4. RED
   denegar por defecto, alcanzada por observar y avisar
   puntos privados, salida con lista de permitidos
   registro de consultas de nombres

5. DATOS
   clasificación al recoger
   claves por ámbito, con inventario de qué protege cada una
   estrategia de borrado para lo inmutable

6. CADENA DE SUMINISTRO
   repositorio interno como único origen
   nombres internos reservados
   embudo de priorización y cadencia de reconstrucción

7. GOBIERNO Y COSTE
   etiquetas impuestas, reparto de compartidos escrito
   coste por unidad publicado
   presupuesto por equipo con aviso por previsión
   regla de apagado por falta de dueño

8. EVIDENCIA
   consultas guardadas para cada control
```

Y las preguntas cuya respuesta hay que escribir:

```
¿cuántas cuentas hay? ¿coincide con la facturación?
¿desde dónde se llega a qué, partiendo de cada punto de entrada?
¿cuántos permisos concedidos no se usan?
¿qué destinos externos usa cada carga?
¿dónde acaban los registros y la telemetría?
¿qué proporción del gasto no está atribuida?
¿qué controles preventivos hay y cuándo se probó cada uno?
```

Las pruebas negativas de la parte 11, que son la entrega más valiosa:

- intentar crear un recurso sin etiquetas de dueño
- intentar crear algo en una región no permitida
- intentar desactivar el registro de auditoría
- intentar borrar una copia de seguridad
- intentar hacer público un almacén
- intentar alcanzar producción desde un entorno inferior
- intentar sacar datos a un destino externo no declarado
- pedir un acceso temporal y cronometrarlo
- usar el acceso de emergencia y comprobar que avisa
- publicar un paquete con el nombre de uno interno
- dejar un recurso sin dueño 30 días

Y una advertencia sobre las once: si alguna pasa, el control no existe aunque esté configurado.

3. Calificación de las cinco predicciones

Predicción 1: «la ley dominante será la 16; más de la mitad de los ejemplos serán controles implantados y luego rodeados, no controles ausentes».

```
veredicto: PARCIALMENTE EQUIVOCADA
```

El reparto real de los problemas tratados en las clases 133 a 143:

```
controles implantados y rodeados o desactivados               8
  filtro de aplicación en aviso 14 meses            135
  revisión trimestral aprobada entera               134
  rotación no ejecutada por exigir reinicio         137
  validación de certificados desactivada            136
  apagado de entornos desactivado la primera semana 143
  cuentas compartidas y accesos alternativos        133
  corrección automática revertida                   139
  compromiso firmado y no revisado                  142

controles AUSENTES o cosas que nadie paró                     9
  9 cuentas fuera de todo inventario                139
  almacén público 19 meses                          139
  salida a un ex-proveedor durante 11 meses         135
  firma de avisos sin validar 14 meses              140
  registros y telemetría fuera de la región         141
  sin cifrado interno entre servicios               136
  nombres internos sin reservar                     138
  compromiso sin dueño 14 meses                     142
  reglas que no detectaban nada                     139
```

Ni la mitad. Y el error es informativo: la ley 16 apareció mucho y no dominó, porque en esta parte la mayoría de los hallazgos graves fueron cosas que nadie había puesto o nadie había parado, no controles esquivados.

Predicción 2: «el gasto mayor no será el que nadie optimizaba: será capacidad para un pico que ya no ocurre o datos que nadie lee. Y el mayor ahorro individual superará la suma de los tres siguientes».

```
primera parte:  ACERTADA, y en las dos formas previstas a la vez
segunda parte:  EQUIVOCADA

partida mayor      compromiso para una campaña que dejó de
                   celebrarse hace 14 meses         24 % de la factura
partida segunda    telemetría que nadie consultaba  16 %

mayor ahorro individual (compromiso)            5.700 €/mes
suma de los tres siguientes                     8.340 €/mes
→ no lo supera
```

Y lo que la segunda parte no vio: el ahorro estaba repartido, no concentrado. Doce medidas de entre 900 y 5.700 euros produjeron más que cualquiera de ellas.

Predicción 3: «la ley 19 reaparecerá en forma financiera: un mecanismo automático ocultando un problema de coste durante meses».

```
veredicto: ACERTADA en el mecanismo, EQUIVOCADA en la duración

lo ocurrido   una consulta sin índice; el autoescalado la absorbió
              sin caída, sin alerta y sin queja
              coste: 2.660 € en 14 días
lo predicho   meses
```

Y el motivo del error es un acierto de la parte anterior: la detección diaria de desviación, montada en la clase 142, redujo a catorce días lo que sin ella habrían sido meses.

Predicción 4: «lo más difícil será la atribución, y la respuesta recurrente será el catálogo».

```
veredicto: ACERTADA, y se quedó corta

la atribución bloqueó tres clases enteras
  139   3 semanas antes de poder repartir un solo hallazgo
  141   5 sistemas con datos personales fuera del inventario
  142   69 % del gasto sin dueño
```

Y lo que la predicción no vio: el catálogo no bastaba. Hizo falta añadirle un control preventivo —etiquetas impuestas en la creación— porque un catálogo se llena con buena voluntad y la buena voluntad caduca.

Predicción 5: «seguridad y coste resultarán ser el mismo problema: recursos que existen sin dueño y sin que nadie sepa por qué».

```
veredicto: ACERTADA, y demostrable con los MISMOS sucesos

suceso                                  seguridad        coste
```

9 cuentas fuera del inventario	almacén público	gasto sin atribuir
salida al ex-proveedor, 11 meses	fuga de datos	tráfico pagado
recursos huérfanos	superficie	1.140 €/mes
permisos sin usar (85 %)	alcance de un compromiso	—
compromiso sin dueño, 14 meses	—	9.800 €/mes

Cinco sucesos que aparecen en las dos columnas o que tienen la misma causa. Era la predicción que podía salir del revés, y salió confirmada.

4. La ley 20, el recuento y la hipótesis de la parte 12

```
LEY 20
  La falta de dueño es la causa común de la fuga y del despilfarro.
  Lo que no tiene responsable no se apaga, no se corrige y no se retira;
  y cada mes que sigue existiendo suma riesgo y factura.
```

Sus cinco apariciones en esta parte:

```
clase 139   9 cuentas sin dueño; una con un almacén público 19 meses
clase 135   un proceso enviando informes a un ex-proveedor 11 meses
clase 141   5 sistemas con datos personales fuera del inventario
clase 142   compromiso de 9.800 €/mes sin revisar durante 14 meses
clase 134   85 % de los permisos concedidos y nunca usados
```

Y lo que añade al cuestionario:

```
¿quién es el dueño de esto, con nombre?
¿qué pasa si esa persona se va?
¿qué lo apagaría si dejara de hacer falta?
¿cuánto tiempo puede existir sin que nadie lo mire?
```

La cuarta es la que más rinde, porque tiene respuesta numérica y se puede vigilar.

Y su relación con las dos leyes vecinas:

```
ley 13   algo deja de funcionar y no da error
ley 19   algo funciona demasiado bien y tapa un problema
ley 20   algo existe y no es de nadie
```

Recuento tras la parte 11:

```
ley 13  el bucle que no corre no da error                       22
ley 15  una señal con demasiados elementos deja de ser señal    20
ley 16  un control que estorba acaba desactivado o rodeado      18
ley 14  las decisiones de creación son irreversibles            11
ley 11  lo que entra en un sistema de solo-añadir se queda       9
ley 20  lo que no tiene dueño no se apaga ni se corrige          5
        NUEVA en esta parte
ley 19  lo que compensa un fallo lo vuelve invisible             6
ley 18  lo asíncrono traslada la garantía, no la elimina         5
ley 17  la medida que se vuelve objetivo se alcanza sin mejorar  6
```

La hipótesis de la parte 12. La parte siguiente cambia el sujeto a la arquitectura: requisitos, límites, división en servicios, consistencia, replicación, contratos y multi-inquilino. La predicción, escrita para poder desmentirla:

```
1. La parte 12 no introducirá mecanismos nuevos: NOMBRARÁ y ordenará
   decisiones que las partes 05 a 11 ya tomaron sin saberlo.
   → predigo que más de la mitad de sus clases formalizarán algo
     ya hecho: la consistencia por operación (109, 110, 111),
     la resiliencia (130), los contratos (115, 118), el aislamiento
     por inquilino (136, 118)

2. La ley dominante será la 14 —las decisiones de creación son
   irreversibles—, porque la arquitectura ES un conjunto de decisiones
   de creación.
   → y predigo que tendrá más apariciones en la parte 12 que en
     cualquier otra parte por separado (el máximo hasta ahora son 3)

3. La clase más difícil será la de dividir el sistema, y la conclusión
   honesta será que la división la deciden la propiedad de los datos y
   los equipos, NO la tecnología.
   → es decir, que la clase 147 determina la 148, y no al revés

4. El hallazgo recurrente será que la arquitectura documentada NO
   coincide con el sistema real, como ya ocurrió con las 18 dependencias
```

```
   no documentadas de la clase 124 y las 35 conexiones de la 135.
   → y predigo que el único artefacto que sobrevive es el que se escribe
     en el momento de decidir, no el diagrama que alguien mantiene

5. Y la predicción que puede salir del revés: «monolito o microservicios»
   resultará ser la decisión MENOS importante de la parte, y las
   consecuentes serán la propiedad de los datos, la consistencia por
   operación y el contrato.
```

Y lo que se anota para calificar sin trampa:

```
lo que ya sabemos    que la documentación diverge, medido dos veces
lo que creemos       que la división es un problema de datos y de equipos
lo que no sabemos    si la quinta es cierta o si la forma de dividir
                     domina de verdad los resultados
```

Ejemplo trabajado

CloudShop rehace su base con todo lo de la parte 11 aplicado desde el principio, y la somete a las once pruebas negativas. Después, el recuento de la parte y las cifras con las que se califican las predicciones.

La estructura resultante.

```
cuentas antes                                                  23
  de ellas, sin dueño                                           9
cuentas después                                                21
  por equipo y entorno                                         15
  de servicios comunes                                          6
    registro y auditoría (nadie más puede escribir)
    red compartida
    seguridad y detección
    repositorio interno de paquetes
    identidad
    facturación y presupuestos
```

Y el alta de un proyecto nuevo, que es lo que decide si la base se usa:

```
tiempo de alta de un proyecto                 6 días         14 min
qué trae montado                              nada       identidad, red,
                                                         registro, etiquetas,
                                                         presupuesto, detección
equipos que pidieron excepción a la base        —              1
  → un servicio heredado; documentada con fecha
```

Las once pruebas negativas.

```
 1. crear un recurso sin etiquetas          rechazado            ✓
 2. crear en una región no permitida        rechazado            ✓
 3. desactivar el registro de auditoría     rechazado            ✓
 4. borrar una copia de seguridad           rechazado            ✓
 5. hacer público un almacén                rechazado            ✓
 6. alcanzar producción desde dev           bloqueado            ✓
 7. sacar datos a un destino no declarado   bloqueado            ✓
 8. pedir acceso temporal, cronometrado     87 s                 ✓
 9. usar el acceso de emergencia            avisó a 4 personas   ✓
                                            revisión abierta     ✓
10. publicar un paquete con nombre interno  el nombre está
                                            reservado            ✓
11. dejar un recurso sin dueño 30 días      apagado con aviso    ✓
```

Once de once. Y en la primera ronda fueron ocho de once:

```
falló la 3   el registro se podía desactivar desde la cuenta de seguridad
             → la política no cubría la propia cuenta de seguridad
falló la 7   una carga tenía una excepción de salida caducada y activa
             → la caducidad no revocaba, solo avisaba
falló la 9   el aviso llegaba a un buzón de un equipo disuelto
             → el mismo hallazgo de las clases 131 y 134
```

Las tres se corrigieron antes de dar por terminada la base. Ninguna se habría descubierto revisando la configuración.

El recuento de la parte 11.

```
claves de larga duración                       14             0
puntos de entrada con alcance total             1             0
puntos desde los que se obtiene otra credencial 4             0
```

```
servicios alcanzables desde uno comprometido  14 de 15        2 de 15
cuentas fuera de inventario                      9              0
almacenes públicos con datos                     1              0
destinos externos no declarados              sin límite         0
permisos concedidos                          8.940          2.100
sin usar nunca                                 85 %           9 %
acceso permanente humano a producción           19             0
secretos existentes                            159            29
entregados por variable de entorno             141             0
hallazgos de vulnerabilidad totales          47.180          2.400
que superan el embudo                           61              1
hallazgos de configuración de prioridad
inmediata                                      103             4
recursos sin dueño                            4.180            0
servicios fuera de la región exigida         5 de 9          0 de 9
tiempo de respuesta a un borrado             9 días          40 min
controles con evidencia consultable          12 de 61       58 de 61
factura mensual                             41.200 €       18.520 €
coste por pedido                             0,229 €        0,057 €
gasto atribuido                                31 %           94 %
controles preventivos con prueba negativa       0              11
```

Las cifras de la calificación.

```
PREDICCIÓN 1 — ley 16 dominante, más de la mitad
  controles rodeados                          8
  controles ausentes o nadie los paró         9
  → no es más de la mitad: EQUIVOCADA

PREDICCIÓN 2 — partida mayor y ahorro concentrado
  partida mayor: compromiso para un pico que ya no ocurre  24 %  ✓
  segunda: telemetría que nadie lee                        16 %  ✓
  mayor ahorro 5.700 € frente a suma de tres siguientes 8.340 €  ✗

PREDICCIÓN 3 — ley 19 en forma financiera
  ocurrió: 2.660 € en 14 días por una consulta sin índice   ✓
  duración predicha: meses                                  ✗

PREDICCIÓN 4 — atribución como problema central
  bloqueó 3 clases; 69 % del gasto sin dueño                ✓
  y el catálogo NO bastó: hizo falta control preventivo     (matiz)

PREDICCIÓN 5 — seguridad y coste, el mismo problema
  5 sucesos aparecen en las dos columnas                    ✓
```

Lo que la parte 11 no resolvió, dicho con claridad.

```
el empleado con acceso legítimo que exporta datos antes de irse
  → detección por volumen, y no prevención                clase 140
los metadatos que salen de la región igualmente          clase 141
la comparabilidad de las cifras de emisiones             clase 143
y el riesgo del compromiso a tres años: sigue siendo una apuesta
```

Las cuatro se documentaron como límites, no como pendientes.

La conclusión que cierra la parte 11: la base pasó las once pruebas negativas a la tercera intentona, y las tres que fallaron al principio —el registro desactivable desde la propia cuenta de seguridad, una excepción caducada que seguía activa y un aviso a un equipo disuelto— no se habrían encontrado revisando la configuración. Y de todos los hallazgos de la parte, los cinco más caros tenían la misma causa: nueve cuentas sin dueño, un proceso que nadie paró, cinco sistemas fuera del inventario, un contrato sin responsable y el 85 % de los permisos concedidos y jamás usados. La seguridad y el coste resultaron ser el mismo problema, y ese problema se llama no saber de quién es cada cosa.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-11-security-governance-finops/144-proyecto-landing-zone-con-guardrails/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 landing-zone-gobernada 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 landing-zone-gobernada 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 base es tan restrictiva que todo el mundo pide excepciones	Ley 16: la excepción se convierte en el camino normal	Haz que la base sea el camino más rápido: alta de proyecto en minutos con todo montado, y mide cuántas excepciones se piden.
Los controles están configurados y no impiden nada	Nunca se han probado en negativo, y algunos no cubren las cuentas de servicio	Ejecuta las once pruebas negativas, incluidas las que atacan a las cuentas de seguridad y registro.
Una excepción caducada sigue funcionando	La caducidad avisa pero no revoca	Que la caducidad revoque de verdad, y compruébalo con una excepción de prueba.
La base se montó a mano y no se puede reproducir ni auditar	No está declarada como infraestructura	Declárala desde el primer día y corrígela en el origen, no en los recursos.
Se cambia la estructura de cuentas al segundo año	Es una de las tres decisiones irreversibles y se tomó sin pensarla	Decide estructura, dominio de identidad y esquema de etiquetado antes que nada; el resto se puede corregir después.
Se dan por buenas las predicciones sin comprobarlas	Calificar solo lo que salió bien convierte el aprendizaje en opinión	Publica el veredicto de cada predicción con su evidencia, incluidas la que falló y la que se quedó corta.

■ Seguridad, ética y costo

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

■ Preguntas de comprobación

1. ¿Qué tres decisiones de una zona de aterrizaje son irreversibles en la práctica?
2. ¿Por qué una base demasiado restrictiva acaba produciendo el efecto contrario?
3. ¿En qué se equivocó la predicción de que la ley 16 dominaría la parte 11?
4. ¿Qué dice la ley 20 y en qué se diferencia de las leyes 13 y 19?
5. ¿Qué predice la hipótesis de la parte 12 sobre la ley dominante y sobre la división en servicios?

Referencias

- AWS (2025). Organizing your environment using multiple accounts — estructura, controles y cuentas de servicio. <https://docs.aws.amazon.com/whitepapers/latest/organizing-your-aws-environment/>
- Google Cloud (2025). Landing zone design — jerarquía, identidad, red y controles de la organización. <https://cloud.google.com/architecture/landing-zones>
- Microsoft (2025). Cloud Adoption Framework: landing zones — áreas de diseño y su orden. <https://learn.microsoft.com/azure/cloud-adoption-framework/ready/landing-zone/>
- CIS (2025). Foundations benchmarks — controles mínimos de una base de nube. <https://www.cisecurity.org/cis-benchmarks>
- FinOps Foundation (2025). Account and tagging strategy — atribución impuesta desde la estructura. <https://www.finops.org/framework/capabilities/allocation/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 11 en PDF · Manual integral

Anterior	Índice	Siguiente
← 143 · Optimización de costo, capacidad y sostenibilidad	Parte 11 · Programa	145 · Requisitos, restricciones y atributos de calidad →

Evaluación — 144 Proyecto: landing zone con guardrails

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 landing-zone-gobernada 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.