Multi-Cloud Engineering Program

Parte 13 — Multi-cloud, híbrido, migración y recuperación

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	13 de 23
Clases	157–168
Evaluaciones	12
Horas estimadas	52

Índice

Parte 13 — Multi-cloud, híbrido, migración y recuperación

← Parte 12 · Índice completo · Parte 14 →

Descargar: Esta parte en PDF · Manual integral

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

Diseñar continuidad y portabilidad sin ocultar complejidad, latencia ni egress.

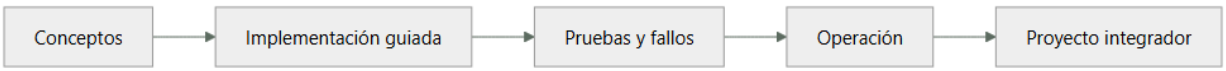
Resultados de la parte

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

Multi-cloud es una decisión de negocio y riesgo; duplicar cada componente entre proveedores rara vez es la forma más confiable o económica de lograrla.

Secuencia



Clases

ID	Clase	Laboratorio	Horas
157	Motivaciones y anti-patrones de multi-cloud	decision	4
158	Portabilidad, capas de abstracción y lock-in	architecture	4
159	Federación de identidad entre nubes	iam	4
160	Conectividad, tránsito, DNS y service discovery	network	4
161	Replicación de datos, soberanía y costos de egress	data	4
162	Observabilidad y operación entre proveedores	observability	4
163	Terraform multi-provider y separación de estados	iac	4
164	Flotas Kubernetes y políticas comunes	kubernetes	4
165	Nube híbrida, edge y conectividad privada	hybrid	4
166	Backup, RTO, RPO y patrones de disaster recovery	reliability	4
167	Las 7R de migración y oleadas	migration	4
168	Proyecto: continuidad activa-pasiva entre nubes	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

- Cloud Strategy — Gregor Hohpe.
- Seeking SRE — Blank-Edelman.
- Enterprise Integration Patterns — Hohpe y Woolf.

157 — Motivaciones y anti-patrones de multi-cloud

Parte: 13 — Multi-cloud, híbrido, migración y recuperación Nivel: experto · Horas estimadas: 4 Laboratorio: decision · Estado: EXECUTABLECORE

Propósito

Decidir si hay que usar varios proveedores y para qué, sometiendo cada motivo declarado a la pregunta de la clase 145: ¿qué pasa si no?. La clase sostiene que el motivo más citado —evitar la dependencia de un proveedor— es el más débil, que el que más justifica es el que menos se menciona, y que la discusión mejora enormemente al dejar de hablar de «multi-nube» y empezar a hablar de cinco niveles distintos, cuyo coste crece muchísimo más deprisa que su beneficio.

Resultados de aprendizaje

Al finalizar podrás:

1. Interrogar cada motivo hasta saber qué se pierde si no se hace.
2. Distinguir dependencia del proveedor de coste de salida, y medir el segundo.
3. Elegir el nivel de multi-nube en vez de discutir el concepto.
4. Enumerar lo que cuesta cada nivel, incluido lo que casi nadie cuenta.
5. Reconocer los cuatro antipatrones, empezando por el mínimo común denominador.

Conceptos centrales

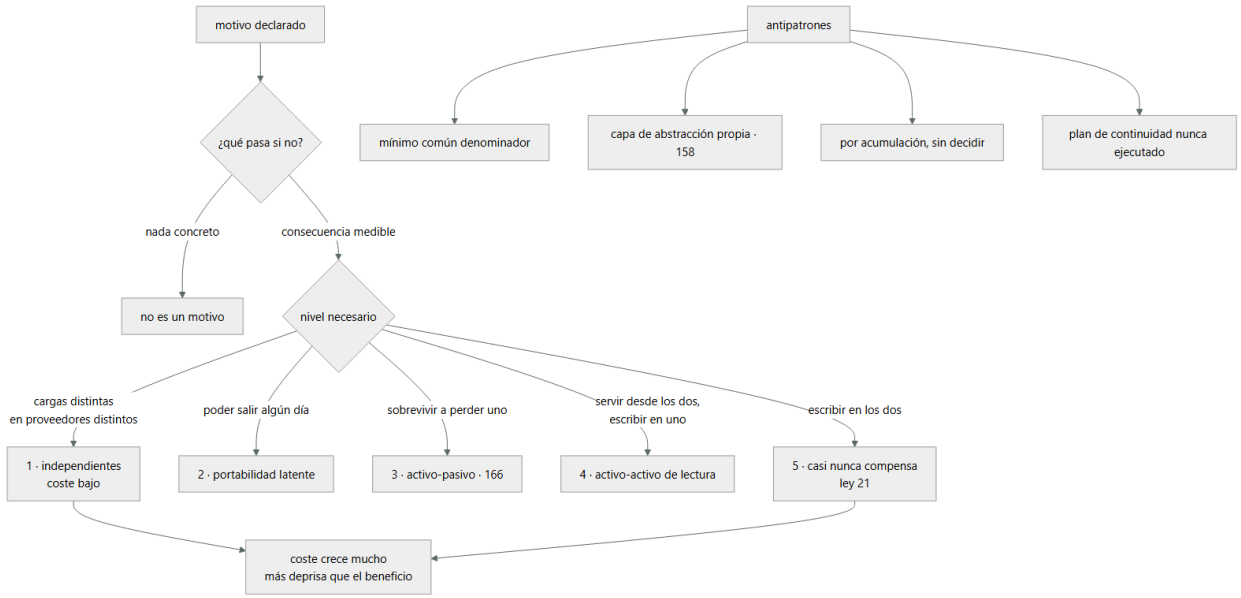
Concepto	Comprensión verificable
coste de salida	Lo que costaría dejar un proveedor: tiempo, dinero y riesgo. Es la magnitud útil; «dependencia» no se puede medir.
nivel de multi-nube	Grado concreto: cargas independientes, portabilidad latente, activo-pasivo, activo-activo de lectura o de escritura.
mínimo común denominador	Usar solo lo que existe en todos los proveedores. Se paga el precio de varios y se obtiene el peor de todos.
multi-nube por acumulación	Estar en varios proveedores sin haberlo decidido, por adquisiciones o por elección de cada equipo.
capacidad exclusiva	Servicio que solo ofrece un proveedor y que aporta una ventaja real. Es uno de los pocos motivos que resisten.
portabilidad latente	Poder salir sin estar fuera: la carga se puede desplegar en otro proveedor, pero solo corre en uno.

Modelo mental

Multi-cloud es una decisión de negocio y riesgo; duplicar cada componente entre proveedores rara vez es la forma más confiable o económica de lograrla.

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. Interrogar los motivos

Los ocho motivos que se declaran en la práctica, sometidos a la pregunta de la clase 145:

1. «EVITAR LA DEPENDENCIA DE UN PROVEEDOR»
   el más citado y el más débil
   ¿qué pasa si no? nada, hasta que haya que salir
   coste de hacerlo se paga TODOS LOS DÍAS
   coste de no hacerlo se paga UNA VEZ, si llega el caso
   → y casi siempre la comparación no se hace

2. «SOBREVIVIR A LA CAÍDA DE UN PROVEEDOR»
   suena fuerte y no resiste la aritmética: ver apartado segundo

3. «TENER FUERZA EN LA NEGOCIACIÓN»
   real, y es un motivo comercial, no técnico
   → y solo funciona si la portabilidad es CREÍBLE, que es cara

4. NORMATIVA O SOBERANÍA
   un cliente o una norma exigen que ciertos datos estén en un sitio
   ¿qué pasa si no? no se puede operar, o no se firma el contrato
   → es el motivo más sólido de la lista clase 141

5. CAPACIDAD QUE SOLO TIENE UN PROVEEDOR
   un servicio concreto que aporta ventaja real
   ¿qué pasa si no? se construye a mano y cuesta más
   → sólido, y suele resolverse en el nivel 1

6. EXIGENCIA DE UN CLIENTE
   «nuestros datos, en este proveedor»
   → sólido, y es una decisión comercial

7. ADQUISICIONES O HISTORIA
   ya se está en dos proveedores porque se compró una empresa
   → no es un motivo: es un HECHO, y hay que decidir qué hacer con él

8. LATENCIA O CERCANÍA A LOS DATOS
   los clientes o los datos están donde uno de ellos está mejor
   → sólido cuando hay cifras de latencia detrás

Y el recuento honesto:

motivos que resisten la pregunta	4, 5, 6 y 8
motivos que no	1 y 2
motivo comercial, no técnico	3
no es un motivo	7

Y la observación incómoda: el más citado es el que menos resiste, y el que más resiste —la normativa— rara vez aparece en la primera conversación.

Sobre la dependencia, que merece reformularse:

```
«dependencia» no se puede medir; «coste de salida» sí

coste de salida de un servicio de cómputo         semanas
coste de salida de una base gestionada            meses
coste de salida de un servicio propietario
  de datos o de aprendizaje                       trimestres
coste de salida de la identidad y la red          el mayor de todos
```

Y con esa tabla la conversación cambia:

```
no es «¿estamos atados?»
es «¿cuánto costaría salir, y es proporcionado a lo que ganamos
   estando aquí?»
→ y si la respuesta es que sí, la dependencia es una decisión,
  no un accidente
```

2. La aritmética de la disponibilidad

El segundo motivo merece su propio apartado porque suena convincente y casi siempre es falso.

La comparación correcta no es «un proveedor frente a dos», sino:

```
dos regiones del mismo proveedor    frente a    dos proveedores
```

Y los datos que hay que poner encima de la mesa:

```
fallos que afectan a UNA región                    frecuentes
fallos que afectan a UN proveedor ENTERO           muy raros
fallos que afectan a un servicio concreto
  en varias regiones                               ocurren
```

Y el coste de cada opción:

```
DOS REGIONES, UN PROVEEDOR
  identidad, red y herramientas comunes
  replicación gestionada por el propio proveedor
  un solo conjunto de conocimientos y de guardia

DOS PROVEEDORES
  identidad federada entre ellos                     clase 159
  conectividad y resolución de nombres entre ellos   clase 160
  replicación propia, y su coste de salida de datos  clase 161
  observabilidad unificada                           clase 162
  infraestructura declarada con dos proveedores      clase 163
  dos conjuntos de conocimientos y dos guardias
  y el mínimo común denominador                      apartado cuarto
```

Y el efecto que casi nadie cuenta:

```
un sistema repartido entre dos proveedores tiene MÁS modos de fallo
que uno bien hecho en dos regiones de uno
→ la conectividad entre nubes, la federación de identidad y la
  replicación propia son componentes nuevos que pueden fallar
→ y suelen fallar más que el proveedor entero
```

Y la conclusión práctica, que hay que decir sin rodeos:

```
si el objetivo es disponibilidad, casi siempre se consigue antes,
mejor y más barato con dos regiones del mismo proveedor

el multi-nube por disponibilidad se justifica cuando
  una norma exige no depender de un solo proveedor, o
  el contrato con un cliente lo exige
→ es decir, vuelve a ser el motivo 4 o el 6, no el 2
```

Y el caso que sí conviene tener cubierto, y es distinto:

```
el proveedor deja de prestar el servicio, sube el precio de forma
inaceptable o hay un conflicto contractual
→ eso no se resuelve con activo-activo: se resuelve con
  PORTABILIDAD LATENTE y con un coste de salida conocido
```

3. Cinco niveles, no un concepto

La discusión mejora al dejar de hablar de «multi-nube» y hablar de niveles concretos:

```
NIVEL 1 · CARGAS INDEPENDIENTES
  cada carga vive entera en un proveedor, elegido por su motivo
  el análisis en uno, la tienda en otro, el correo en un tercero
  + coste marginal bajo; no hace falta portabilidad
  + resuelve los motivos 4, 5, 6 y 8
  – hace falta identidad y observabilidad comunes

NIVEL 2 · PORTABILIDAD LATENTE
  la carga PODRÍA desplegarse en otro; solo corre en uno
  + acota el coste de salida sin pagarlo a diario
  – exige disciplina: evitar servicios propietarios en el camino crítico
  – y la portabilidad no probada no existe

NIVEL 3 · ACTIVO-PASIVO
  copia en frío o en caliente en el otro proveedor         clase 166
  + sobrevive a perder uno, con un plazo declarado
  – replicación continua, coste de salida de datos y ensayos

NIVEL 4 · ACTIVO-ACTIVO DE LECTURA
  se sirve desde los dos; se escribe solo en uno
  + latencia y tolerancia mejores
  – todo lo del nivel 3, más el enrutado y la coherencia de lectura

NIVEL 5 · ACTIVO-ACTIVO DE ESCRITURA
  se escribe en los dos
  – conflictos garantizados                                clase 149
  – y el dato tiene dos escritores                         ley 21
  → casi nunca compensa; ver el apartado siguiente
```

Y la relación entre coste y beneficio:

```
nivel     coste relativo    lo que aporta de más
  1            1×          casi todo lo que la gente quiere
  2          1,2×          capacidad de salir
  3          2-3×          sobrevivir a perder un proveedor
  4          3-4×          latencia y tolerancia
  5          5-8×          escritura en los dos lados
```

Y el hallazgo que ordena la parte: la mayoría de los motivos legítimos se resuelven en el nivel 1, que es el más barato y el que casi nadie llama multi-nube.

Y sobre el nivel 5, la conclusión que esta parte va a sostener:

```
escribir en dos proveedores a la vez significa que un dato
tiene dos escritores
→ conflictos, resolución de conflictos y todo lo de la clase 149
→ y la latencia entre nubes hace inviable coordinar

compensa solo cuando
  los datos se pueden PARTIR por región o por cliente, de modo que
  cada dato tenga un solo escritor aunque el sistema esté en los dos
→ y entonces no es activo-activo de escritura: es nivel 1 partido
```

4. Los cuatro antipatrones

```
1. MÍNIMO COMÚN DENOMINADOR
   usar solo lo que existe en todos los proveedores
   → nada de bases gestionadas específicas, nada de servicios
     propietarios, nada de lo que hace barato usar una nube
   → se paga el precio de varios proveedores y se obtiene el peor
     de todos
   → y el equipo acaba operando a mano lo que estaba resuelto

2. LA CAPA DE ABSTRACCIÓN PROPIA
   una biblioteca interna que oculta las diferencias
   → acaba siendo un producto que hay que mantener
   → con menos funciones que cualquiera de los originales
   → y una dependencia nueva de la que sí es difícil salir
   → es la materia de la clase 158

3. MULTI-NUBE POR ACUMULACIÓN
   se está en tres proveedores porque cada equipo eligió el suyo
   o porque se compró una empresa
   → se pagan todos los costes y no se obtiene ningún beneficio,
     porque nada es portable ni redundante
   → es lo más común en la práctica

4. EL PLAN DE CONTINUIDAD QUE NUNCA SE HA EJECUTADO
   existe un segundo proveedor «por si acaso»
   → nunca se ha conmutado, nadie sabe cuánto tarda y el procedimiento
     está desactualizado
   → ley 13: lo que no se ejecuta no da ningún error
   → y en el momento de usarlo, no funciona
```

Y el tercero merece una precisión, porque suele ser el punto de partida real:

```
estar en varios proveedores sin haberlo decidido NO es un fracaso:
es una situación
→ lo que hay que decidir es qué se hace con ella
   consolidar en uno, o
   declarar el nivel 1 y ordenar identidad y observabilidad
→ lo que no vale es dejarlo sin decidir y llamarlo estrategia
```

Y el método completo de esta clase, en cinco pasos:

```
1. escribir el motivo, con nombre y quién lo sostiene
2. preguntarle «¿qué pasa si no?» y anotar la consecuencia medible
3. si no hay consecuencia, retirar el motivo
4. para los que quedan, elegir el NIVEL mínimo que los satisface
5. medir el coste de salida actual y decidir si es proporcionado
```

Y las cifras que conviene tener antes de decidir:

```
qué proporción del gasto está en cada proveedor
qué servicios propietarios se usan en el camino crítico
qué costaría, en semanas, mover cada carga
cuánto costaría la salida de datos de mover el histórico   clase 161
y cuántas personas saben operar cada proveedor
```

La última suele ser la que decide: con un equipo pequeño, dos proveedores significan que la mitad de la gente no puede intervenir en la mitad del sistema.

Y la lista de comprobación de la clase:

- ■ cada motivo está escrito, con quién lo sostiene
- ■ cada motivo ha pasado por «¿qué pasa si no?»
- ■ los que no tienen consecuencia medible se han retirado
- ■ se ha comparado dos regiones frente a dos proveedores, con cifras
- ■ está elegido el NIVEL, no el concepto
- ■ el nivel elegido es el mínimo que satisface los motivos
- ■ el coste de salida está estimado por carga, en semanas
- ■ no se está usando el mínimo común denominador sin decidirlo
- ■ no se ha construido una capa de abstracción propia
- ■ si hay plan de continuidad, se ha ejecutado alguna vez
- ■ está contado cuántas personas saben operar cada proveedor

Y el cierre que enlaza con la clase siguiente: el motivo más citado se apoya en una idea que conviene examinar despacio —que la portabilidad se consigue con una capa de abstracción—. Qué es portable de verdad, qué cuesta

serlo y por qué la abstracción propia suele ser peor que la dependencia que evita es la materia de la clase 158.

Ejemplo trabajado

La dirección de CloudShop pide «una estrategia multi-nube» tras una caída de su proveedor. El ejercicio consiste en escribir los motivos y someterlos a la pregunta. De ocho, sobreviven tres, y ninguno es el que originó la petición.

Los ocho motivos, tal como se declararon.

```
1. «no queremos depender de un solo proveedor»          dirección
2. «que una caída del proveedor no nos pare»            dirección
3. «para negociar mejor el contrato»                    finanzas
4. «tres clientes exigen datos en su región»            comercial
5. «el servicio de análisis del otro es mejor»          datos
6. «un cliente exige otro proveedor»                    comercial
7. «heredamos una cuenta de la empresa que compramos»   hecho
8. «latencia para los clientes de otra zona»            producto
```

La interrogación.

```
MOTIVO 2, el que originó todo
  ¿qué pasa si no?   la caída que lo motivó duró 41 min y afectó
                     a UNA REGIÓN, no al proveedor
  ¿se resolvía con multi-nube?   no: con una segunda región
  coste de la segunda región                          +2.900 €/mes
  coste del nivel 3 entre proveedores                 +11.400 €/mes
  → RETIRADO como motivo de multi-nube; se abre un trabajo
    de segunda región

MOTIVO 1
  ¿qué pasa si no?   «estaríamos atados»
  reformulado como coste de salida:
    cómputo y contenedores                    3-4 semanas
    bases gestionadas                         3-4 meses
    servicio de análisis propietario          6+ meses
    identidad y red                           el mayor
  ¿es proporcionado a lo que ganamos?   sí, para todo salvo el análisis
  → RETIRADO como motivo general; queda una acción concreta:
    acotar el coste de salida del análisis

MOTIVO 3
  ¿qué pasa si no?   se negocia peor
  ¿cuánto?           el descuento actual es del 34 %; el equipo de
                     compras estima 3-5 puntos más con alternativa creíble
  coste de la alternativa creíble (nivel 2)           +1.800 €/mes
  beneficio estimado                                  ~700 €/mes
  → NO COMPENSA con el gasto actual; se revisa si el gasto se dobla

MOTIVOS 4 y 6
  ¿qué pasa si no?   no se firman tres contratos, y uno se pierde
  valor de esos contratos                            ~240.000 €/año
  → SOBREVIVEN                                        clase 141

MOTIVO 5
  ¿qué pasa si no?   se construye a mano; estimación 4 meses de dos
                     personas y peor resultado
  → SOBREVIVE

MOTIVO 7
  no es un motivo: es una cuenta con 6 servicios heredados
  → decisión aparte: consolidar o declarar nivel 1

MOTIVO 8
  ¿qué pasa si no?   latencia de 240 ms para el 4 % de los clientes
  ¿lo resuelve otro proveedor?   no: lo resuelve una región del mismo
  → RETIRADO; se abre un trabajo de región adicional
```

El recuento.

```
motivos declarados                                      8
sobreviven a la pregunta                                3   (4, 5, 6)
retirados por no tener consecuencia medible             2   (1, 8)
retirados porque se resuelven mejor de otra forma       1   (2)
```

```
no compensa con los números actuales                          1   (3)
no era un motivo                                              1   (7)
```

Tres de ocho. Y el que originó la petición —sobrevivir a una caída— resultó ser un problema de regiones, no de proveedores.

El nivel elegido.

Los tres motivos supervivientes se satisfacen todos en el nivel 1:

```
motivo 4   los datos de tres clientes en su región del otro proveedor
           → una carga independiente allí
motivo 5   el análisis en el otro proveedor
           → una carga independiente allí
motivo 6   un cliente entero servido desde el otro proveedor
           → una celda independiente allí               clase 151

nivel elegido                                          1
nivel que la propuesta inicial pedía                   4
diferencia de coste estimada                  11.400 €/mes
```

Y lo que sí hubo que construir aunque el nivel sea el más barato:

```
identidad federada entre los dos                       clase 159
observabilidad unificada                               clase 162
infraestructura declarada con dos proveedores          clase 163
y nada de red entre nubes: las cargas son independientes  clase 160
```

La última línea es la que abarata el nivel 1: si las cargas no se hablan, no hace falta conectarlas.

La cuenta heredada, decidida.

```
servicios heredados en el tercer proveedor                    6
  2 los usa un cliente que sigue activo    → se mantienen, nivel 1
  3 no los usa nadie desde hace 14 meses   → se apagan  ley 20
  1 es una base con datos históricos       → se migra al lago

coste mensual antes                                    1.900 €
coste mensual después                                    310 €
cuentas activas                                            3 → 2
```

Los antipatrones, revisados.

```
MÍNIMO COMÚN DENOMINADOR
  la propuesta inicial incluía «usar solo Kubernetes y PostgreSQL
  en ambos, sin servicios gestionados específicos»
  coste estimado de operar bases no gestionadas    2 personas a tiempo
                                                   parcial, permanente
  → descartado: con nivel 1 no hace falta

CAPA DE ABSTRACCIÓN PROPIA
  se propuso una biblioteca interna para el almacenamiento de objetos
  → descartada; ver clase 158

POR ACUMULACIÓN
  era la situación real: tres proveedores sin decisión
  → resuelta arriba

PLAN NUNCA EJECUTADO
  existía un documento de continuidad de 2023
  ¿se ha ejecutado alguna vez?                    no
  ¿alguien sabe cuánto tardaría?                  no
  → se convierte en el trabajo de las clases 166 y 168
```

Las cifras que decidieron.

```
gasto por proveedor        principal 94 % · segundo 4 % · tercero 2 %
servicios propietarios en el camino crítico                    2
coste de salida de la carga principal, estimado          6 semanas
coste de salida del análisis                             6 meses
personas que saben operar el proveedor principal               9
personas que saben operar el segundo                           2
```

Y la última fila fue la que más pesó en la decisión del nivel:

```
con 2 personas capaces de operar el segundo proveedor,
un nivel 3 o 4 significaría que 7 de 9 no pueden intervenir
```

```
en la mitad del sistema durante un incidente
→ y eso empeora la disponibilidad, que era el motivo original
```

A los doce meses.

proveedores	3	2
proveedores decididos a propósito	0	2
motivos escritos e interrogados	0	8
motivos vivos	—	3
nivel de multi-nube	sin decidir	1
cargas en el segundo proveedor	1	3
conectividad entre nubes	no había	sigue sin haber
coste del tercer proveedor	1.900 €	0 €
personas capaces de operar el segundo	2	5
coste de salida documentado por carga	no	sí
plan de continuidad ejecutado alguna vez	no	pendiente (clase 168)

La lección que esta clase abre para la parte 13: de ocho motivos declarados, tres sobrevivieron a la pregunta, y el que puso en marcha toda la conversación —sobrevivir a una caída del proveedor— resultó ser un problema de regiones que costaba cuatro veces menos resolver dentro del mismo proveedor. Y la cifra que más pesó no fue ninguna de coste: fue que solo dos personas de nueve sabían operar el segundo proveedor, lo que convertía cualquier nivel avanzado en una amenaza para la disponibilidad que se quería mejorar.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-13-multicloud-hybrid-disaster-recovery/157-motivaciones-y-anti-patrones-de-multi-cloud/lab.py
```

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

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

Evidencia esperada

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

Reto verificable

Construye decision-multicloud 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 decide usar varios proveedores sin poder explicar qué se pierde si no	El motivo es un adjetivo, no una consecuencia medible	Aplica la pregunta de la clase 145 a cada motivo y retira los que no tengan consecuencia con cifra.
Se busca disponibilidad con dos proveedores y el sistema falla más	Se comparó un proveedor con dos, en vez de dos regiones con dos proveedores; y se añadieron componentes nuevos que fallan	Haz la comparación correcta con las frecuencias reales de fallo y elige regiones salvo que una norma o un contrato exijan otra cosa.
Se pagan varios proveedores y se usa lo peor de todos	Mínimo común denominador: solo lo que existe en todos	Usa servicios gestionados específicos en el nivel 1 y acota el coste de salida en vez de renunciar a ellos.
Se está en tres proveedores y no se obtiene ninguna ventaja	Multi-nube por acumulación, sin decisión	Decide: consolidar o declarar nivel 1 con identidad y observabilidad comunes; lo que no vale es dejarlo sin decidir.
Existe un segundo proveedor por si acaso y nadie sabe si funcionaría	Ley 13: el plan nunca se ha ejecutado	Ejecuta una conmutación real y mide; sin eso, el plan no existe.
La discusión sobre multi-nube no avanza	Se discute el concepto en vez del nivel concreto	Elige entre los cinco niveles el mínimo que satisface los motivos supervivientes.

Seguridad, ética y costo

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

Preguntas de comprobación

1. ¿Cuál es el motivo más citado y por qué es el más débil?
2. ¿Cuál es la comparación correcta cuando el objetivo es disponibilidad?
3. ¿Cuáles son los cinco niveles y cuál resuelve la mayoría de los motivos legítimos?
4. ¿Por qué el mínimo común denominador cuesta más y da menos?
5. ¿Qué magnitud sustituye a la palabra dependencia y cómo se mide?

Referencias

- Gartner (2025). Multicloud strategies: motivations and pitfalls — motivos declarados frente a resultados. <https://www.gartner.com/en/information-technology>
- Google Cloud (2025). Multicloud and hybrid architecture patterns — niveles concretos y su coste. <https://cloud.google.com/architecture/hybrid-multicloud-patterns>
- AWS (2025). Multi-Region fundamentals — comparación entre regiones y proveedores para disponibilidad. <https://docs.aws.amazon.com/whitepapers/latest/aws-multi-region-fundamentals/>
- Fowler, M. (2025). Exit cost over lock-in — sustituir dependencia por coste de salida medible. <https://martinfowler.com/bliki/>
- CNCF (2025). Multicloud interoperability — qué es común entre proveedores y qué no. <https://www.cncf.io/reports/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 13 en PDF · Manual integral

Evaluación — 157 Motivaciones y anti-patrones de multi-cloud

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 decision-multicloud 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.

158 — Portabilidad, capas de abstracción y lock-in

Parte: 13 — Multi-cloud, híbrido, migración y recuperación Nivel: experto · Horas estimadas: 4 Laboratorio: architecture · Estado: EXECUTABLECORE

Propósito

Examinar la idea en la que se apoya el motivo más citado de la clase anterior: que la portabilidad se consigue construyendo una capa que oculte las diferencias. La clase separa qué es portable de verdad y qué no, capa por capa, muestra por qué la abstracción propia suele acabar siendo una dependencia peor que la que evitaba, y propone la alternativa que sí funciona: usar las interfaces que ya son estándar, empujar lo específico al borde y conocer el coste de salida en semanas en vez de intentar anularlo.

Resultados de aprendizaje

Al finalizar podrás:

1. Situar cada capa del sistema por su portabilidad real.
2. Reconocer cuándo una capa de abstracción propia es una trampa y cuándo un adaptador es correcto.
3. Aprovechar las interfaces que ya son comunes, que dan portabilidad gratis.
4. Medir el coste de salida por carga, en semanas y en euros.
5. Comprobar la portabilidad periódicamente, porque la que no se prueba no existe.

Conceptos centrales

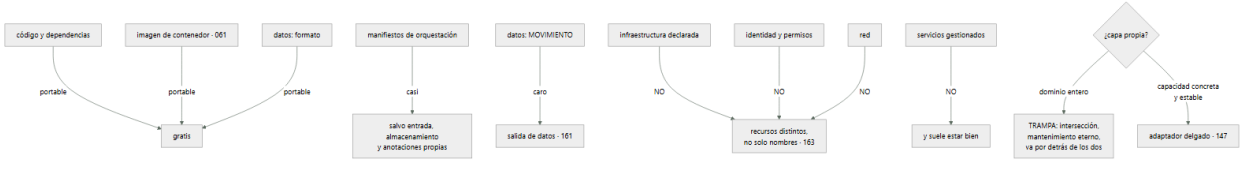
Concepto	Comprensión verificable
portabilidad	Capacidad de ejecutar lo mismo en otro proveedor con un esfuerzo acotado. No es binaria: se mide por capas y en semanas.
interfaz común	Especificación que varios proveedores implementan. Da portabilidad sin construir nada.
capa de abstracción propia	Biblioteca interna que oculta las diferencias entre proveedores. Suele exponer la intersección de funciones y hay que mantenerla siempre.
adaptador delgado	Traducción de una capacidad concreta y estable en la frontera. Es correcto; la abstracción de un dominio entero no.
coste de salida	Semanas de trabajo y euros de movimiento de datos para dejar un proveedor. Se estima por carga y se actualiza cada año.
prueba de portabilidad	Construir y arrancar en el segundo proveedor de forma periódica, aunque no se sirva tráfico desde allí.

Modelo mental

Multi-cloud es una decisión de negocio y riesgo; duplicar cada componente entre proveedores rara vez es la forma más confiable o económica de lograrla.

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 es portable, capa por capa

La portabilidad no es una propiedad del sistema: es distinta en cada capa, y el coste de salida vive en las de abajo.

```
CÓDIGO Y DEPENDENCIAS                    portable
  salvo bibliotecas propietarias del proveedor

IMAGEN DE CONTENEDOR                     portable              clase 061
  la especificación es común; se ejecuta en cualquier sitio

MANIFIESTOS DE ORQUESTACIÓN              casi portable
  la API es común y hay diferencias reales:
    entrada y balanceo, con anotaciones propias
    clases de almacenamiento
    integración con la identidad del proveedor
    autoescalado de nodos

FORMATO DE LOS DATOS                     portable              clase 112
  columnar, JSON, SQL estándar

MOVIMIENTO DE LOS DATOS                  caro                  clase 161
  el formato es portable y mover cien teratabytes cuesta dinero
  y tiempo

INFRAESTRUCTURA DECLARADA                NO portable           clase 163
  no son nombres distintos: son recursos con modelos distintos

IDENTIDAD Y PERMISOS                     NO portable           clase 159
  los modelos de permisos no se corresponden entre sí

RED                                      NO portable           clase 160

SERVICIOS GESTIONADOS                    NO portable
  colas, bases, aprendizaje, análisis
  → y renunciar a ellos cuesta más de lo que la portabilidad vale
```

Y la lectura de la tabla:

```
las capas de arriba son portables y NADIE se preocupa por ellas
las de abajo no lo son y son donde está el coste de salida
→ y una capa de abstracción propia solo actúa sobre las de en medio
```

Y una precisión sobre los manifiestos, que es donde más gente se lleva sorpresas:

```
«es Kubernetes, es portable»
  el 80-90 % de los manifiestos, sí
  el resto son las cuatro diferencias de arriba
  → y ese resto es exactamente lo que conecta la carga con el proveedor
```

Y sobre los servicios gestionados, la posición honesta de este programa:

```
renunciar a ellos para ser portable significa
  operar bases, colas y almacenes a mano              partes 09 y 10
  con un equipo que ya tiene su trabajo
→ el coste diario supera casi siempre al coste de salida ocasional
→ la respuesta correcta no es renunciar: es SABER lo que costaría salir
```

2. La trampa de la capa propia

La propuesta aparece siempre: una biblioteca interna que oculte las diferencias entre proveedores.

Y falla por cinco motivos que se repiten:

```
1. EXPONE LA INTERSECCIÓN
   solo puede ofrecer lo que existe en todos
   → es el mínimo común denominador de la clase 157, en código

2. HAY QUE MANTENERLA SIEMPRE
   los proveedores publican funciones nuevas cada semana
   → y cada una exige decidir si se expone y cómo
   → sin comunidad que lo haga por ti
```

```
3. VA POR DETRÁS DE LOS DOS
   quien la usa no puede aprovechar lo nuevo hasta que se añada

4. LA DEPURACIÓN PASA POR ELLA
   un error del proveedor llega envuelto en tu abstracción
   → y la documentación del proveedor deja de aplicar

5. ES UNA DEPENDENCIA DE LA QUE SÍ ES DIFÍCIL SALIR
   nadie más la conoce, no hay quien la mantenga si el autor se va
   → se evitó depender de un proveedor con miles de ingenieros
     y se depende de una biblioteca con uno
```

Y la excepción legítima, que conviene distinguir con precisión:

```
ADAPTADOR DELGADO, correcto
  una capacidad concreta, estable y pequeña
  «guardar y leer un objeto», «publicar un mensaje»
  con la interfaz de TU dominio, no la del proveedor
  → es la capa de traducción de la clase 147
  → cabe en una página y no crece

ABSTRACCIÓN DE UN DOMINIO ENTERO, trampa
  «una API común para cualquier base de datos»
  «un modelo unificado de identidad»
  → crece sin fin y expone la intersección
```

Y la prueba práctica para distinguirlas:

```
¿cuántos métodos tiene?
  menos de diez y estables      → adaptador
  decenas y creciendo           → abstracción

¿qué pasa cuando el proveedor añade algo útil?
  no me afecta, no lo uso       → adaptador
  tengo que exponerlo           → abstracción

¿podría reescribirla en dos días si hiciera falta?
  sí                            → adaptador
```

Y una forma sana de conseguir el efecto sin la trampa, que es la de la clase 147:

```
el núcleo del sistema define lo que NECESITA
  «necesito guardar un documento y recuperarlo por su identificador»
y hay una implementación por proveedor, cada una usando lo suyo
  a fondo, sin limitarse a la intersección
→ el núcleo no conoce a ningún proveedor
→ y cada implementación puede usar lo mejor de cada uno
```

La diferencia con la abstracción es sutil y decisiva: la interfaz la dicta tu dominio, no el mínimo común de los proveedores.

3. Portabilidad que sale gratis

Hay interfaces que varios proveedores implementan, y usarlas da portabilidad sin construir nada:

```
imágenes de contenedor                  especificación común
API de orquestación                     común, con las salvedades ya dichas
SQL estándar                            en lo que no sean extensiones
protocolo de almacenamiento de objetos  implementado por varios
telemetría                              formato común              clase 124
identidad basada en testigos estándar   clase 159
colas con protocolos abiertos           según el caso
formatos de datos columnares            clase 112
```

Y dos advertencias importantes sobre esta lista:

```
«compatible con» no es «igual que»
  una implementación compatible del protocolo de objetos suele
  cubrir las operaciones básicas y no el control de acceso,
  las políticas de ciclo de vida ni la consistencia fina

y lo que rompe es siempre lo mismo
  el comportamiento en los bordes: errores, límites, consistencia
  → la clase 153 ya lo dijo: el contrato es más que la forma
```

Y las tres prácticas que dan portabilidad real sin capa propia:

```
1. LO ESPECÍFICO, EN EL BORDE
   el núcleo no menciona a ningún proveedor
   → clase 147, aplicada al proveedor en vez de a un tercero

2. INFRAESTRUCTURA POR PROVEEDOR, NO ABSTRACTA
   módulos separados con la misma interfaz de entrada y salida
   → un módulo «base de datos de pedidos» por proveedor
   → y NO un módulo que intente parametrizar los dos    clase 163

3. CONOCER EL COSTE DE SALIDA Y ACEPTARLO
   la opción más barata y la que menos se elige
```

Y la tercera merece defenderse, porque suena a rendición y no lo es:

```
usar a fondo un servicio gestionado y saber que salir cuesta
seis semanas es una DECISIÓN
lo que no vale es no saberlo
```

Y un caso concreto donde la portabilidad sí conviene pagarse:

```
lo que está en el camino crítico Y es difícil de sustituir
  la base de datos principal
  el sistema de identidad
  el almacén de datos históricos
→ ahí conviene usar interfaces estándar aunque cueste algo

lo que no está en el camino crítico o es fácil de sustituir
  colas, cachés, funciones, herramientas
→ ahí conviene usar lo mejor de cada proveedor
```

4. Medir el coste de salida y probarlo

El método, que cabe en una tabla y se actualiza una vez al año:

```
para cada carga
  1. listar lo específico del proveedor que usa
  2. para cada elemento, estimar el trabajo de sustituirlo, en semanas
  3. estimar el volumen de datos a mover y su coste       clase 161
  4. sumar, y añadir un factor por lo que no se ha previsto
  5. anotar la fecha y quién lo estimó
```

Y un ejemplo de la forma que tiene:

```
carga: servicio de pedidos
  cómputo en contenedores            1 semana
  base gestionada                    6 semanas  (esquema + migración)
  cola gestionada                    2 semanas
  almacén de objetos                 1 semana
  identidad y permisos               3 semanas
  red y entrada                      2 semanas
  observabilidad                     1 semana
                                   ■■■■■■■■■■
                                    16 semanas
  datos a mover                      4,1 TB → coste de salida  ~370 €
  factor por imprevistos             ×1,4
                                   ■■■■■■■■■■
  coste de salida estimado           22 semanas y ~370 €
```

Y lo que se hace con esa cifra:

```
se compara con lo que se gana estando ahí
se decide si es proporcionada
y se identifica QUÉ elemento domina la suma
  → aquí, la base gestionada: 6 de 16 semanas
  → y si se quiere bajar el coste de salida, se ataca ese
```

La prueba de portabilidad, que es lo que separa una portabilidad real de una supuesta:

```
la portabilidad que no se prueba no existe            ley 13
```

Y la versión más barata y creíble:

```
CONSTRUIR Y ARRANCAR en el segundo proveedor, periódicamente
  no servir tráfico: solo demostrar que construye, arranca,
  pasa sus comprobaciones de salud y responde una petición
  con una copia reducida de datos

coste          unas horas de cómputo al mes
```

```
lo que detecta lo que se ha vuelto no portable sin que nadie lo notara
```

Y lo que aparece siempre la primera vez:

```
una anotación específica en los manifiestos
una clase de almacenamiento que no existe
un servicio gestionado que se coló en el camino crítico
una dependencia de la identidad del proveedor
y un valor por defecto que solo es correcto en el original
```

Y la variante completa, más cara y para cuando el nivel lo exige:

```
desplegar y servir un porcentaje pequeño de tráfico real
→ y esto ya es el nivel 3 o 4 de la clase 157, con su coste
```

Y la lista de comprobación de la clase:

- ■ está identificado qué capa del sistema es portable y cuál no
- ■ el núcleo no menciona a ningún proveedor
- ■ no existe una capa de abstracción propia de un dominio entero
- ■ los adaptadores tienen pocos métodos y son estables
- ■ se usan interfaces comunes donde existen, sabiendo sus límites
- ■ los módulos de infraestructura son por proveedor, no abstractos
- ■ el coste de salida está estimado por carga, en semanas y euros
- ■ está identificado el elemento que domina esa suma
- ■ la estimación tiene fecha y se actualiza cada año
- ■ hay una prueba periódica de construir y arrancar en el segundo
- ■ los servicios propietarios del camino crítico están declarados

Y el cierre que enlaza con la clase siguiente: de todas las capas de la tabla, la que menos se puede portar y la que más problemas causa al operar entre proveedores es la identidad. Cómo se federa, cómo se evita duplicar permisos y por qué es lo primero que hay que montar es la materia de la clase 159.

Ejemplo trabajado

CloudShop, ya en dos proveedores en nivel 1, recibe una propuesta para construir una capa de abstracción interna. El ejercicio consiste en medir primero el coste de salida real y decidir con esa cifra delante.

La propuesta.

```
«una biblioteca interna que abstraiga almacenamiento de objetos,
 colas, secretos y base de datos, para poder cambiar de proveedor»
esfuerzo estimado por quien la proponía                  8 semanas
mantenimiento estimado                                   «poco»
```

Paso 1: medir el coste de salida antes de decidir nada.

```
carga                        semanas    lo que domina
servicio de pedidos            22        base gestionada (6)
catálogo                        9        almacén de objetos (2)
análisis                       26        servicio propietario (14)
identidad y red (transversal)  11        —
observabilidad                  4        —
                             ■■■■■■
                               72 semanas para salir del todo

datos a mover                  118 TB
coste de salida de datos       ~9.400 €                  clase 161
```

Y el desglose por lo que la capa propuesta habría cubierto:

```
lo que la capa abstraería
  almacén de objetos                    3 semanas de 72     4 %
  colas                                 4 semanas de 72     6 %
  secretos                              2 semanas de 72     3 %
  base de datos                         6 semanas de 72     8 %
                                      ■■■■■■■■■■■■■■■■■■■■
                                       15 semanas de 72    21 %

lo que NO abstraería
  identidad y permisos                 11 semanas
  red y entrada                         6 semanas
  servicio propietario de análisis     14 semanas
  esquema y migración de datos         16 semanas
  observabilidad                        4 semanas
```

```
  resto                                   6 semanas
                                        ■■■■■■■■■■■
                                        57 semanas          79 %
```

La capa cubría el 21 % del coste de salida y costaba ocho semanas de construcción más mantenimiento permanente.

Paso 2: la prueba de las tres preguntas.

```
¿cuántos métodos tendría?
  estimados en el diseño                                  64
  y creciendo con cada necesidad nueva            → abstracción

¿qué pasa cuando el proveedor añade algo útil?
  hay que exponerlo o no se puede usar            → abstracción

¿se podría reescribir en dos días?
  no                                              → abstracción
```

Tres de tres. La propuesta se rechazó, y en su lugar:

```
adaptadores delgados donde ya existían por otro motivo
  almacenamiento de objetos     7 métodos, ya existía por la clase 147
  secretos                      3 métodos
  colas                         4 métodos
total                          14 métodos, estables
trabajo adicional              0 semanas: ya estaban
```

Paso 3: el elemento que domina, atacado.

Con la tabla delante, quedó claro dónde estaba el coste:

```
servicio propietario de análisis               14 semanas
esquema y migración de la base                 16 semanas
identidad y permisos                           11 semanas
                                              ■■■■■■■■■■■
                                               41 de 72 semanas
```

Y se tomaron tres decisiones concretas, en vez de una capa genérica:

```
1. ANÁLISIS
   los datos ya viven en el lago en formato columnar     clase 112
   lo propietario es el motor de consulta, no los datos
   → se acotó a que ningún resultado se guarde solo en ese servicio
   coste de salida                    de 14 a 5 semanas

2. BASE DE DATOS
   se usaba una extensión propietaria en 3 consultas
   → se sustituyeron por SQL estándar
   coste de salida                    de 6 a 4 semanas

3. IDENTIDAD
   no se puede portar; se documentó y se aceptó         clase 159
   coste de salida                    11 semanas, sin cambio
```

```
coste de salida total                      72 semanas     58 semanas
trabajo invertido                               —          3 semanas
coste de la capa propuesta                 8 semanas +      —
                                           mantenimiento
cobertura de esa capa                         21 %          —
```

Tres semanas de trabajo dirigido bajaron el coste de salida más que ocho semanas de capa genérica, porque atacaron lo que dominaba la suma.

Paso 4: la prueba de portabilidad, y lo que encontró.

Se montó la prueba barata: construir y arrancar el catálogo en el segundo proveedor una vez al mes.

```
primera ejecución                                    falló
causas
  1. una anotación de entrada específica del proveedor original
  2. una clase de almacenamiento inexistente
  3. el cliente de secretos asumía la identidad del proveedor
  4. un valor por defecto de zona horaria distinto
  5. el registro de imágenes no era accesible desde allí
tiempo en corregirlas                                4 días
```

Y el valor de la prueba se vio en los meses siguientes:

```
ejecuciones en 12 meses                                  12
fallos                                                    5
  de ellos, por algo que se había vuelto no portable
  sin que nadie lo notara                                 5
  ejemplos: una función gestionada añadida al camino crítico,
            una anotación nueva, una dependencia de un servicio
            de mensajería propietario
tiempo medio de corrección                            1 día
```

Y el contraste con el mismo ejercicio en la carga que no tenía prueba:

```
servicio de pedidos, sin prueba periódica
al intentar arrancarlo en el segundo proveedor, 14 meses después
  fallos encontrados                                     23
  tiempo estimado de corrección                  5 semanas
```

Cinco fallos corregidos en un día cada uno, frente a veintitrés acumulados. La portabilidad se pierde poco a poco y solo se conserva si se comprueba.

El caso de «compatible con».

```
el catálogo usaba el protocolo estándar de almacenamiento de objetos
contra una implementación compatible del segundo proveedor

lo que funcionó                              lectura y escritura
lo que NO funcionó
  las políticas de ciclo de vida             formato distinto
  el control de acceso por política de objeto modelo distinto
  las notificaciones al escribir un objeto   no existían
  la consistencia de listado tras escribir   comportamiento distinto
```

Y el último produjo un error real durante la prueba:

```
el proceso escribía un objeto y lo listaba a continuación
en el original, aparecía siempre
en el compatible, a veces no
→ el código dependía de un comportamiento no prometido    clase 153
```

A los doce meses.

```
capa de abstracción propia                 propuesta      no existe
adaptadores delgados                             3              3
métodos en total                                14             14
coste de salida total                    72 semanas     58 semanas
elemento que domina la suma              análisis (14)  esquema (16)
estimación con fecha y responsable               no             sí
prueba de portabilidad                     no había     mensual, 2 cargas
fallos de portabilidad detectados por ella       —          5 / año
servicios propietarios en el camino crítico      4              2
extensiones de SQL propietarias                  3              0
```

La lección que esta clase traslada a la parte 13: la capa de abstracción propuesta cubría el 21 % del coste de salida y habría añadido una dependencia permanente; tres semanas de trabajo dirigido al 57 % que dominaba la suma consiguieron más. Y la portabilidad no se pierde de golpe: se pierde poco a poco, cinco fallos al año en la carga que se probaba y veintitrés acumulados en la que no, que es la ley 13 aplicada a la capacidad de irse.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-13-multicloud-hybrid-disaster-recovery/158-portabilidad-capas-de-abstraccion-y-lock-in/lab.py
```

El laboratorio selecciona el motor de práctica architecture 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-portabilidad 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 diagrama acompañado por decisiones y trade-offs. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye matriz-portabilidad 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 construye una biblioteca interna para abstraer proveedores y acaba limitando a todos	Expone la intersección de funciones y hay que mantenerla siempre	Usa adaptadores delgados de pocos métodos con la interfaz de tu dominio, y una implementación por proveedor que use lo mejor de cada uno.
Se invierte en portabilidad y el coste de salida apenas baja	Se abstrajo lo fácil, que era una fracción pequeña de la suma	Mide el coste de salida por carga, identifica qué elemento domina y ataca ese.
Al intentar desplegar en otro proveedor aparecen decenas de incompatibilidades	La portabilidad se perdió poco a poco y nadie lo comprobaba	Prueba mensual de construir y arrancar en el segundo proveedor, aunque no sirva tráfico.
Una implementación compatible falla en los bordes	«Compatible con» cubre las operaciones básicas, no errores, límites ni consistencia	Comprueba el comportamiento en los bordes y no dependas de lo que no se promete.
Se renuncia a servicios gestionados para ser portable y el equipo se sobrecarga	Se paga a diario el coste de operar para evitar un coste ocasional	Usa servicios gestionados y documenta el coste de salida; la decisión es conocerlo, no anularlo.
Nadie sabe cuánto costaría dejar el proveedor	No se ha estimado nunca	Tabla por carga con semanas por elemento y coste de mover datos, con fecha y responsable, actualizada cada año.

■ 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 capas del sistema son portables y cuáles no, y dónde vive el coste de salida?
2. ¿Por qué falla una capa de abstracción propia y en qué se distingue de un adaptador delgado?
3. ¿Qué significa exactamente que una implementación sea compatible con un protocolo?

4. ¿Cómo se estima el coste de salida y qué se hace con esa cifra?
5. ¿Cuál es la prueba de portabilidad más barata que resulta creíble?

Referencias

- OCI (2025). Image and runtime specifications — portabilidad real de la capa de contenedores. <https://opencontainers.org/>
- CNCF (2025). Cloud native landscape and interoperability — qué interfaces son comunes entre proveedores. <https://landscape.cncf.io/>
- Cockcroft, A. (2025). Lock-in and exit cost — sustituir el debate por una estimación en semanas. <https://adrianco.medium.com/>
- Hexagonal architecture (2025). Ports and adapters — mantener lo específico en el borde. <https://alistair.cockburn.us/hexagonal-architecture/>
- Google Cloud (2025). Portability considerations for hybrid and multicloud — límites prácticos de la portabilidad. <https://cloud.google.com/architecture/hybrid-multicloud-patterns/considerations>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 13 en PDF · Manual integral

Anterior	Índice	Siguiente
← 157 · Motivaciones y anti-patrones de multi-cloud	Parte 13 · Programa	159 · Federación de identidad entre nubes →

Evaluación — 158 Portabilidad, capas de abstracción y lock-in

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-portabilidad 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.

159 — Federación de identidad entre nubes

Parte: 13 — Multi-cloud, híbrido, migración y recuperación Nivel: experto · Horas estimadas: 4 Laboratorio: iam · Estado: EXECUTABLECORE

Propósito

Montar lo primero que hay que montar al usar dos proveedores, y lo que menos se puede portar: la identidad. La clase separa dos problemas que se confunden —las personas y las cargas—, da la solución correcta para cada uno, y se detiene en el detalle que convierte una federación en un agujero: una condición de confianza demasiado amplia. Y afronta con honestidad lo que no se puede unificar: los modelos de permisos de los proveedores no se corresponden, así que no hay un rol común, sino la misma intención implementada dos veces y algo que compruebe que siguen siendo equivalentes.

Resultados de aprendizaje

Al finalizar podrás:

1. Separar la federación de personas de la de cargas.
2. Sustituir claves de larga duración entre nubes por credenciales temporales.
3. Acotar la condición de confianza a lo que debe poder usarla.
4. Reconocer qué diferencias de modelo impiden unificar los permisos.
5. Responder quién hizo qué cuando la misma persona tiene identidades distintas.

Conceptos centrales

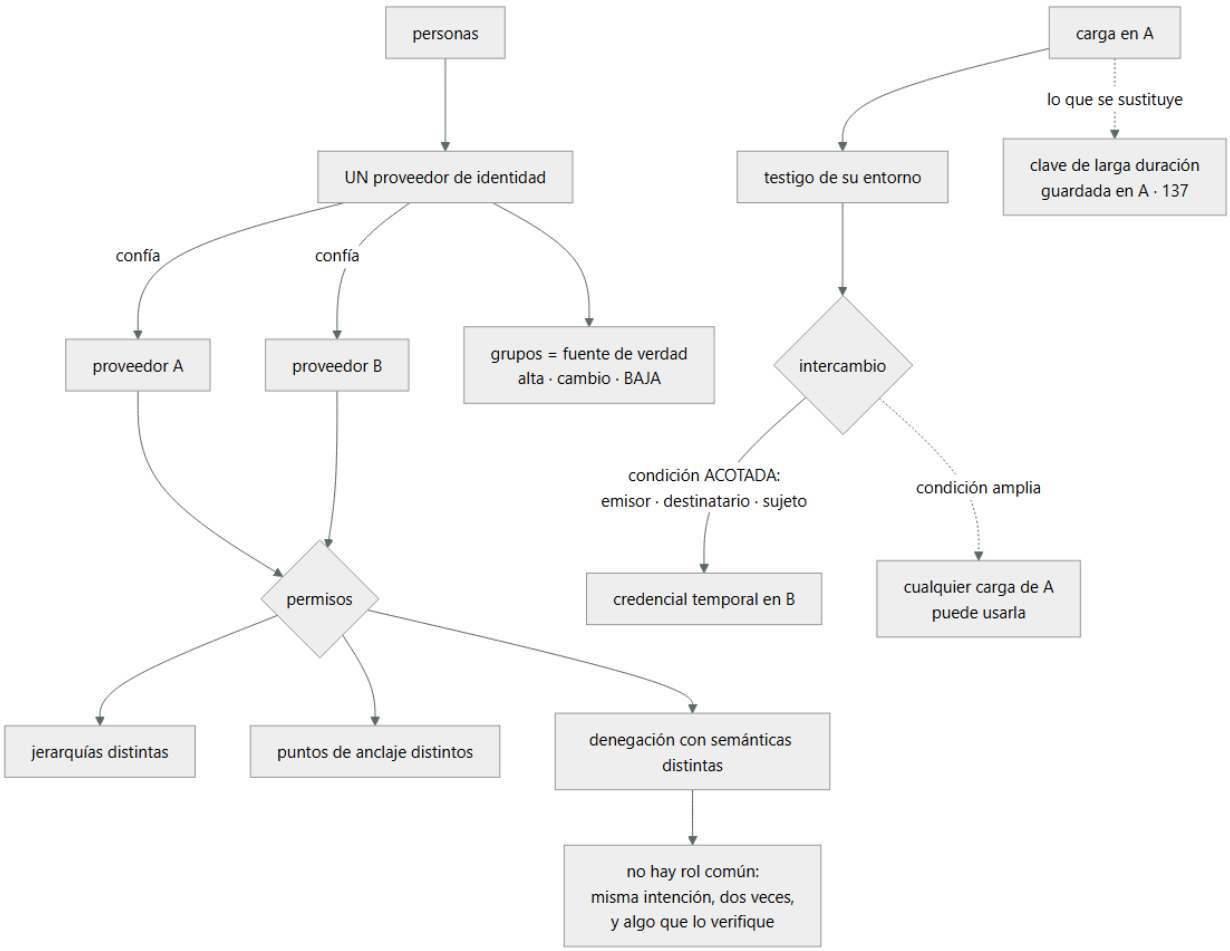
Concepto	Comprensión verificable
proveedor de identidad	Sistema donde viven las personas y sus grupos. Debe haber uno, y los proveedores de nube confían en él.
federación de carga	Una carga demuestra quién es con el testigo de su entorno y obtiene credenciales temporales en otro proveedor. No hay secreto que guardar.
condición de confianza	Qué emisor, qué destinatario y qué sujeto concreto se aceptan. Si es amplia, cualquier carga de ese entorno puede usarla.
intención equivalente	El mismo permiso expresado en dos modelos distintos. No se puede unificar; hay que verificar que coincide.
ciclo de vida de la persona	Alta, cambio y baja. Si no es único para todos los proveedores, alguien se va y queda activo en uno.
correlación de identidades	Saber que dos sujetos distintos en dos proveedores son la misma persona. Sin ella no se puede auditar.

Modelo mental

Multi-cloud es una decisión de negocio y riesgo; duplicar cada componente entre proveedores rara vez es la forma más confiable o económica de lograrla.

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. Dos problemas distintos

Se llaman igual y se resuelven de forma distinta:

```
PERSONAS
  alguien de desarrollo necesita entrar en los dos proveedores
  → problema bien resuelto: un proveedor de identidad y confianza
    desde cada nube

CARGAS
  un servicio que corre en A necesita llamar a un servicio de B
  → problema mal resuelto casi siempre: una clave estática guardada
```

Las personas. La regla es una:

```
UN proveedor de identidad, y los dos proveedores de nube confían en él
→ nadie tiene usuario propio en ninguna nube
→ los grupos del proveedor de identidad se traducen a roles en cada uno
```

Y lo que eso resuelve de golpe:

```
el factor resistente a suplantación se exige una vez     clase 133
la baja de una persona se hace en un sitio
el acceso temporal se concede sobre grupos                clase 134
y las condiciones de acceso —dispositivo, origen— se aplican una vez
```

Y el detalle que decide si funciona:

```
los GRUPOS son la fuente de verdad, no las asignaciones en cada nube
→ si alguien asigna un rol directamente a una persona en un proveedor,
  esa asignación sobrevive a su baja
→ y eso hay que impedirlo con un control preventivo    clase 139
```

Las cargas. Las tres opciones, ordenadas:

```
1. FEDERACIÓN DE CARGA        ← la correcta
   la carga presenta el testigo que le da su entorno en A
   B lo verifica y entrega credenciales temporales
   → no hay secreto que guardar ni rotar              clase 137
   → es el patrón de la clase 098, generalizado entre nubes

2. CLAVE DE LARGA DURACIÓN DE B GUARDADA EN A
   → funciona el primer día y es lo que la clase 137 dedicó
     una clase entera a eliminar
   → y ahora está en un proveedor distinto del que la emitió,
     con lo que el rastro es más largo

3. INTERMEDIARIO QUE EMITE CREDENCIALES
   un servicio propio al que las cargas piden credenciales
   → añade un componente crítico y el problema del arranque
     de confianza                                    clase 137
   → solo si el proveedor no admite federación directa
```

Y el orden de trabajo que se deduce, y que la clase 157 ya anticipó:

```
la identidad es lo PRIMERO que se monta al añadir un proveedor
porque todo lo demás la necesita, y porque montarla después
significa migrar todo lo que se hizo mientras tanto
```

2. La condición de confianza

La federación de carga se apoya en una relación de confianza, y su configuración es donde está el riesgo.

Lo que se declara:

```
EMISOR         quién firma el testigo: el entorno de A
DESTINATARIO   para quién es el testigo: el proveedor B
SUJETO         QUÉ carga concreta puede usarlo
CADUCIDAD      cuánto vale la credencial resultante
```

Y el tercero es el que se configura mal:

```
mal   «acepto cualquier testigo emitido por el clúster de A»
      → cualquier carga de ese clúster puede obtener credenciales en B
      → incluida una comprometida, y las de entornos inferiores

bien  «acepto el testigo cuyo sujeto sea la cuenta de servicio
       pedidos del espacio de nombres produccion del clúster X»
```

Y es exactamente el mismo error que la clase 098 encontró con las canalizaciones: confiar en el emisor sin acotar el sujeto.

Y las comprobaciones que hay que hacer sobre cada relación de confianza:

```
¿el sujeto está acotado a una identidad concreta?
¿los entornos inferiores tienen relaciones distintas de producción?
¿la caducidad de la credencial resultante es corta?
¿los permisos que se obtienen son los mínimos?          clase 134
¿queda registro en los DOS proveedores de cada intercambio?
```

Y una prueba negativa que conviene ejecutar, del catálogo de la clase 131:

```
intentar obtener credenciales de B desde una carga de A
que NO debería poder
→ y comprobar que se rechaza y que queda registro
```

Y dos cautelas más:

```
LA CADUCIDAD DEL TESTIGO NO ES LA DE LA CREDENCIAL
  un testigo de una hora puede producir credenciales de doce
  → hay que acotar las dos

EL TESTIGO ES UN SECRETO MIENTRAS VIVE
  si se filtra en un registro, sirve para obtener credenciales
  → depuración obligatoria                          clase 122
```

Y sobre el sentido de la federación, que conviene decidir explícitamente:

```
¿de A hacia B, de B hacia A, o en los dos sentidos?
→ cada sentido es una relación de confianza más y una superficie más
→ y casi siempre hace falta uno solo
```

3. Lo que no se puede unificar

Aquí está el motivo por el que la identidad es la capa menos portable: los modelos de permisos no se corresponden.

```
JERARQUÍA DE RECURSOS
  los tres proveedores organizan los recursos en árboles distintos,
  con niveles distintos y significados distintos
  → una política aplicada «al nivel de arriba» no significa lo mismo

DÓNDE SE ANCLA LA POLÍTICA
  a la identidad, al recurso, al contenedor de recursos, o a varios
  → y el resultado de combinarlas se calcula de forma distinta

SEMÁNTICA DE LA DENEGACIÓN
  denegación explícita que gana siempre
  políticas de denegación aparte
  restricciones de organización que acotan lo que se puede conceder
  → tres mecanismos con reglas de precedencia distintas    clase 049

CONDICIONES Y ATRIBUTOS
  qué se puede condicionar —origen, etiqueta, hora, dispositivo—
  y cómo se expresa

IDENTIDADES DE CARGA
  una tiene cuentas de servicio, otra roles asumibles, otra identidades
  administradas
```

Y la consecuencia práctica, que hay que aceptar:

```
no existe «un rol común para los dos proveedores»
existe la MISMA INTENCIÓN implementada dos veces
y algo que compruebe que siguen siendo equivalentes
```

Y cómo se hace esa comprobación, que es lo único que evita la deriva:

```
1. escribir la intención en un lenguaje propio y neutro
   «el equipo de pedidos puede leer y escribir sus datos de pedidos,
    y no puede tocar copias de seguridad ni el registro de auditoría»

2. implementarla en cada proveedor con su modelo, a fondo

3. y comprobar el RESULTADO, no la configuración
   ¿puede esta identidad borrar una copia en A? ¿y en B?
   ¿puede leer los datos de otro equipo? ¿en los dos?
```

Y el tercer paso es la clave: comparar configuraciones es imposible; comparar lo que se puede hacer, no. Y se automatiza como las pruebas negativas de la clase 144.

Y lo que sí se puede unificar sin forzar nada:

```
los grupos de personas                          en el proveedor de identidad
las etiquetas de dueño                          clase 142
el catálogo de quién es responsable de qué      clase 095
y las fronteras absolutas, expresadas en cada modelo
  «nadie puede desactivar el registro de auditoría»
  «nadie puede borrar copias»                   clase 139
```

La última línea es la más rentable: son pocas reglas, se expresan en los dos modelos y son las que más daño evitarían.

4. Ciclo de vida, emergencia y auditoría

El ciclo de vida de una persona es donde falla la federación en la práctica:

```
ALTA      se crea en el proveedor de identidad y hereda grupos
CAMBIO    cambia de equipo; sus grupos cambian
BAJA      se desactiva en el proveedor de identidad
```

Y lo que se escapa:

```
asignaciones directas hechas en un proveedor, fuera de los grupos
cuentas locales creadas «temporalmente» en una nube
claves de acceso creadas por esa persona y no revocadas
identidades de carga creadas a su nombre
y accesos a herramientas de terceros con su cuenta
```

Y la medida que lo detecta:

```
personas dadas de baja en el proveedor de identidad
con algo activo en algún proveedor de nube
→ debería ser cero, y la primera vez que se mide nunca lo es
```

El acceso de emergencia, que la clase 134 exigió y que ahora se duplica:

```
no puede depender del proveedor de identidad federado
  porque su caída es justo uno de los casos en que hace falta
y hace falta uno POR PROVEEDOR de nube
  con aviso, revisión obligatoria y rotación tras el uso
y se ensaya una vez al año, en los dos
```

La auditoría entre proveedores, que es el problema que nadie prevé:

```
la misma persona tiene
  un identificador en el proveedor de identidad
  un sujeto distinto en el registro de auditoría de A
  otro sujeto distinto en el de B
→ y la pregunta «¿qué hizo esta persona ayer?» no tiene respuesta
  sin una tabla que los relacione
```

Y lo que hace falta:

```
una correspondencia entre identificadores, mantenida automáticamente
el identificador del proveedor de identidad presente en los registros
  de los dos, cuando el proveedor lo permita
y los registros de auditoría de ambos en un mismo sitio   clase 162
```

Y una comprobación que conviene hacer antes de necesitarla:

```
elegir una persona y reconstruir todo lo que hizo ayer en los dos
proveedores
→ si lleva más de diez minutos, falta la correspondencia
```

Y la lista de comprobación de la clase:

- hay un único proveedor de identidad y los dos confían en él
- nadie tiene usuario local en ninguna nube
- los grupos son la fuente de verdad y las asignaciones directas están impedidas
- no hay claves de larga duración entre proveedores
- cada relación de confianza acota el sujeto, no solo el emisor
- los entornos inferiores tienen relaciones distintas de producción
- la caducidad del testigo y la de la credencial están acotadas
- se ha probado que una carga no autorizada no puede obtener credenciales
- la intención de permisos está escrita en un lenguaje neutro
- se comprueba el RESULTADO en los dos, no la configuración
- las fronteras absolutas están expresadas en los dos modelos
- es cero el número de personas dadas de baja con algo activo
- hay acceso de emergencia por proveedor, ensayado
- existe correspondencia de identificadores para auditar

Y el cierre que enlaza con la clase siguiente: con la identidad resuelta, la siguiente capa que no se puede portar y que decide si dos proveedores pueden trabajar juntos es la red: cómo se alcanzan, cómo se resuelven los nombres y qué cuesta el tránsito. Es la materia de la clase 160.

Ejemplo trabajado

CloudShop federa la identidad entre sus dos proveedores. El ejercicio empieza por inventariar cómo se autentican hoy las cargas y las personas, y encuentra tres cosas que llevaban meses activas.

El inventario de partida.

```
PERSONAS
  proveedor de identidad corporativo                      sí
  usuarios locales en el proveedor principal               4
  usuarios locales en el segundo proveedor                11
  asignaciones de rol directas, fuera de grupos            27

CARGAS
  cargas en A que llaman a B                               3
  cómo se autentican           clave de larga duración de B
                               guardada en el almacén de A
  antigüedad de esas claves                       14 meses
  rotaciones                                              0
```

Y las once cuentas locales del segundo proveedor:

```
creadas «temporalmente» durante la integración inicial
personas que ya no están en la empresa                        3
cuentas sin uso en más de 6 meses                             7
con permisos de administración                                2
```

Tres personas dadas de baja hacía meses seguían con acceso activo al segundo proveedor, porque la baja se hizo en el proveedor de identidad y esas cuentas eran locales.

```
usuarios locales                                 15             0
personas de baja con acceso activo                3             0
asignaciones directas fuera de grupos            27             0
control preventivo que impide crear usuarios
locales                                          no            sí
medida «bajas con algo activo», vigilada         no       semanal, = 0
```

La federación de cargas, y la condición demasiado amplia.

La primera configuración de la relación de confianza:

```
emisor        el clúster de producción de A
destinatario  el proveedor B
sujeto        cualquiera
caducidad     12 horas
```

Y la prueba negativa correspondiente, ejecutada antes de dar por buena la configuración:

```
se desplegó una carga de prueba en el mismo clúster,
en otro espacio de nombres, sin ningún permiso previsto
resultado   obtuvo credenciales válidas en B
            con los permisos del rol de pedidos
```

Cualquier carga del clúster podía actuar como el servicio de pedidos en el otro proveedor.

```
sujeto de la confianza                    cualquiera    cuenta de servicio
                                                        y espacio de nombres
                                                        concretos
relaciones distintas por entorno                no            sí
caducidad de la credencial                    12 h          1 h
prueba negativa ejecutada                       no       trimestral
cargas que pueden obtener credenciales
en B                                       todas (≈180)       3
```

Y las tres claves de larga duración desaparecieron:

```
claves de B guardadas en A                          3 → 0
rotaciones pendientes                               3 → 0
tiempo desde filtración hasta invalidación   indefinido → ≤ 1 h
```

Lo que no se pudo unificar.

Se intentó primero definir «un rol equivalente» y se abandonó al chocar con los modelos:

```
intención        «el equipo de pedidos gestiona sus datos y no toca
                  copias ni auditoría»

en el proveedor A   3 políticas, ancladas a la identidad y al proyecto,
                    más una restricción de organización
en el proveedor B   2 políticas, una de ellas de denegación explícita,
                    ancladas al recurso y a la unidad organizativa

líneas de configuración equivalentes                    0
```

Y se pasó al método del apartado tercero: comprobar el resultado.

```
batería de preguntas, ejecutada contra los dos proveedores
  ¿puede leer sus datos de pedidos?              sí / sí     ✓
  ¿puede escribirlos?                            sí / sí     ✓
  ¿puede leer datos de otro equipo?              no / no     ✓
  ¿puede borrar una copia de seguridad?          no / SÍ     ✗
  ¿puede desactivar el registro de auditoría?    no / no     ✓
  ¿puede crear identidades nuevas?               no / SÍ     ✗
  ¿puede salir a internet sin restricción?       no / SÍ     ✗
```

Tres divergencias, todas en el segundo proveedor, todas por un permiso heredado que en el primero estaba cortado por una restricción de organización que allí no existía.

```
preguntas equivalentes                        4 de 7         7 de 7
fronteras absolutas expresadas en los dos       1              5
comprobación automática del resultado           no       en la canalización
divergencias detectadas después                 —          2 en 8 meses
```

Y las cinco fronteras absolutas que se expresaron en los dos modelos:

```
nadie borra copias de seguridad
nadie desactiva el registro de auditoría
nadie crea identidades con permisos amplios
nadie usa regiones fuera de las autorizadas
nadie hace público un almacén de objetos
```

El acceso de emergencia, duplicado.

```
accesos de emergencia                            1              2
dependen del proveedor de identidad federado     sí             no
ensayo anual                                  no se había      hecho
hallazgo del primer ensayo
  el del segundo proveedor no funcionaba: la credencial se había
  creado con una cuenta local que se eliminó en la limpieza
```

La auditoría entre proveedores.

La prueba del apartado cuarto, ejecutada:

```
«reconstruye todo lo que hizo esta persona ayer, en los dos»
  primer intento                                      2 h 40
  causa    el sujeto en A es un correo, en B es un identificador
           interno, y en el proveedor de identidad es otro
```

```
correspondencia de identificadores           no había      mantenida
                                                           automáticamente
registros de auditoría en un solo sitio          no              sí  clase 162
tiempo de reconstruir la actividad de
una persona                                   2 h 40          4 min
```

El orden en que se hizo, y por qué importó.

```
semana 1-2   personas: proveedor único, sin usuarios locales
semana 3-4   cargas: federación y retirada de las 3 claves
semana 5     fronteras absolutas en los dos modelos
semana 6     comprobación de resultado en la canalización
semana 7-8   correspondencia de identificadores y auditoría común
```

Y la razón del orden es la del apartado primero:

```
todo lo que se despliegue en el segundo proveedor a partir de ahora
nace con identidad federada
→ las cargas que se desplegaron ANTES de la semana 4 hubo que migrarlas
→ y fueron 3; si se hubiera montado seis meses después, habrían sido 20
```

A los ocho meses.

```
usuarios locales en las nubes                    15              0
personas de baja con acceso activo                3              0
claves de larga duración entre nubes              3              0
cargas que podían obtener credenciales en B     ≈180             3
relaciones de confianza con sujeto acotado    0 de 1         3 de 3
preguntas de permiso equivalentes             4 de 7         7 de 7
fronteras absolutas en los dos modelos            1              5
accesos de emergencia ensayados                   0              2
tiempo de auditar a una persona                2 h 40          4 min
```

La lección que esta clase traslada a la parte 13: la federación funcionó a la primera y la configuración inicial permitía que cualquiera de las ciento ochenta cargas del clúster actuara como el servicio de pedidos en el otro proveedor, exactamente el mismo error que la clase 098 encontró con las canalizaciones: acotar el emisor y no el sujeto. Y el intento de unificar los permisos se abandonó al descubrir que no había ni una línea de configuración equivalente entre los dos modelos; lo que sí funcionó fue comprobar el resultado con siete preguntas, que encontraron tres divergencias que ninguna comparación de configuraciones habría revelado.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-13-multicloud-hybrid-disaster-recovery/159-federacion-de-identidad-entre-nubes/
```

```
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 federacion-multicloud 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 federacion-multicloud 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
Alguien deja la empresa y sigue teniendo acceso a una de las nubes	Hay usuarios locales o asignaciones directas fuera de los grupos	Un solo proveedor de identidad, prohibición preventiva de usuarios locales y medida semanal de bajas con algo activo.
Cualquier carga de un clúster puede actuar como otra en el segundo proveedor	La condición de confianza acota el emisor pero no el sujeto	Acota a la identidad concreta, separa relaciones por entorno y comprueba con una prueba negativa que otra carga no lo consigue.
Hay claves de larga duración de un proveedor guardadas en otro	Se resolvió la llamada entre nubes con un secreto en vez de con federación	Federación de carga: testigo del entorno intercambiado por credenciales temporales.
Se intenta definir un rol común entre proveedores y no encaja	Los modelos de jerarquía, anclaje y denegación no se corresponden	Escribe la intención en lenguaje neutro, implementa en cada modelo a fondo y comprueba el resultado con una batería de preguntas.
No se puede responder qué hizo una persona en los dos proveedores	Los identificadores de sujeto son distintos y no hay correspondencia	Mantén una correspondencia automática y reúne los registros de auditoría en un solo sitio.
El acceso de emergencia del segundo proveedor no funciona cuando hace falta	Depende del proveedor de identidad federado o de una cuenta que se eliminó	Uno por proveedor, independiente de la federación, con ensayo anual.

■ 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 identidad es lo primero que se monta al añadir un proveedor?
2. ¿Qué tres opciones hay para que una carga de A llame a B y cuál es la correcta?
3. ¿Qué se declara en una relación de confianza y cuál de esos elementos se configura mal?
4. ¿Por qué no existe un rol común entre proveedores y qué se hace en su lugar?
5. ¿Qué hace falta para poder auditar a una persona en dos proveedores?

Referencias

• OpenID Foundation (2025). OpenID Connect Core — base de la federación de personas entre sistemas. <https://openid.net/specs/openid-connect-core-10.html>

• AWS (2025). Identity providers and OIDC federation for workloads — intercambio de testigo por credenciales temporales. <https://docs.aws.amazon.com/IAM/latest/UserGuide/idrolesprovidersoidc.html>

• Google Cloud (2025). Workload identity federation — condiciones de sujeto y acotación de la confianza. <https://cloud.google.com/iam/docs/workload-identity-federation>

• Microsoft (2025). Workload identity federation in Entra ID — federación entre proveedores sin secretos. <https://learn.microsoft.com/entra/workload-id/workload-identity-federation>

• SPIFFE (2025). Identity across trust domains — identidad de carga entre dominios distintos. <https://spiffe.io/docs/latest/spiffe-about/overview/>

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

Descargar: Parte 13 en PDF · Manual integral

Anterior	Índice	Siguiente
← 158 · Portabilidad, capas de abstracción y lock-in	Parte 13 · Programa	160 · Conectividad, tránsito, DNS y service discovery →

Evaluación — 159 Federación de identidad entre nubes

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 federacion-multicloud 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.

160 — Conectividad, tránsito, DNS y service discovery

Parte: 13 — Multi-cloud, híbrido, migración y recuperación Nivel: experto · Horas estimadas: 4 Laboratorio: network · Estado: EXECUTABLECORE

Propósito

Conectar dos proveedores, empezando por la pregunta que ahorra más trabajo que ninguna otra de esta parte: ¿de verdad tienen que hablarse?. La clase muestra que la mayor parte del coste y de la complejidad viene de conectar cosas que no lo necesitaban; que el cable casi nunca es el problema y el direccionamiento sí; y que extender el descubrimiento interno de un proveedor al otro es un error que se paga durante años. Y termina con lo que ocurre cuando el enlace se cae, que es cuando se descubre si las dos partes eran independientes o solo lo parecían.

Resultados de aprendizaje

Al finalizar podrás:

1. Decidir si hace falta conectividad entre proveedores, antes de elegir cómo.
2. Elegir el mecanismo de enlace por coste, previsibilidad y número de extremos.
3. Planificar el direccionamiento para que no haya solapes, que es irreversible.
4. Resolver nombres entre proveedores sin que un resolutor sea un punto único.
5. Exponer una frontera estable en vez de extender el descubrimiento interno.

Conceptos centrales

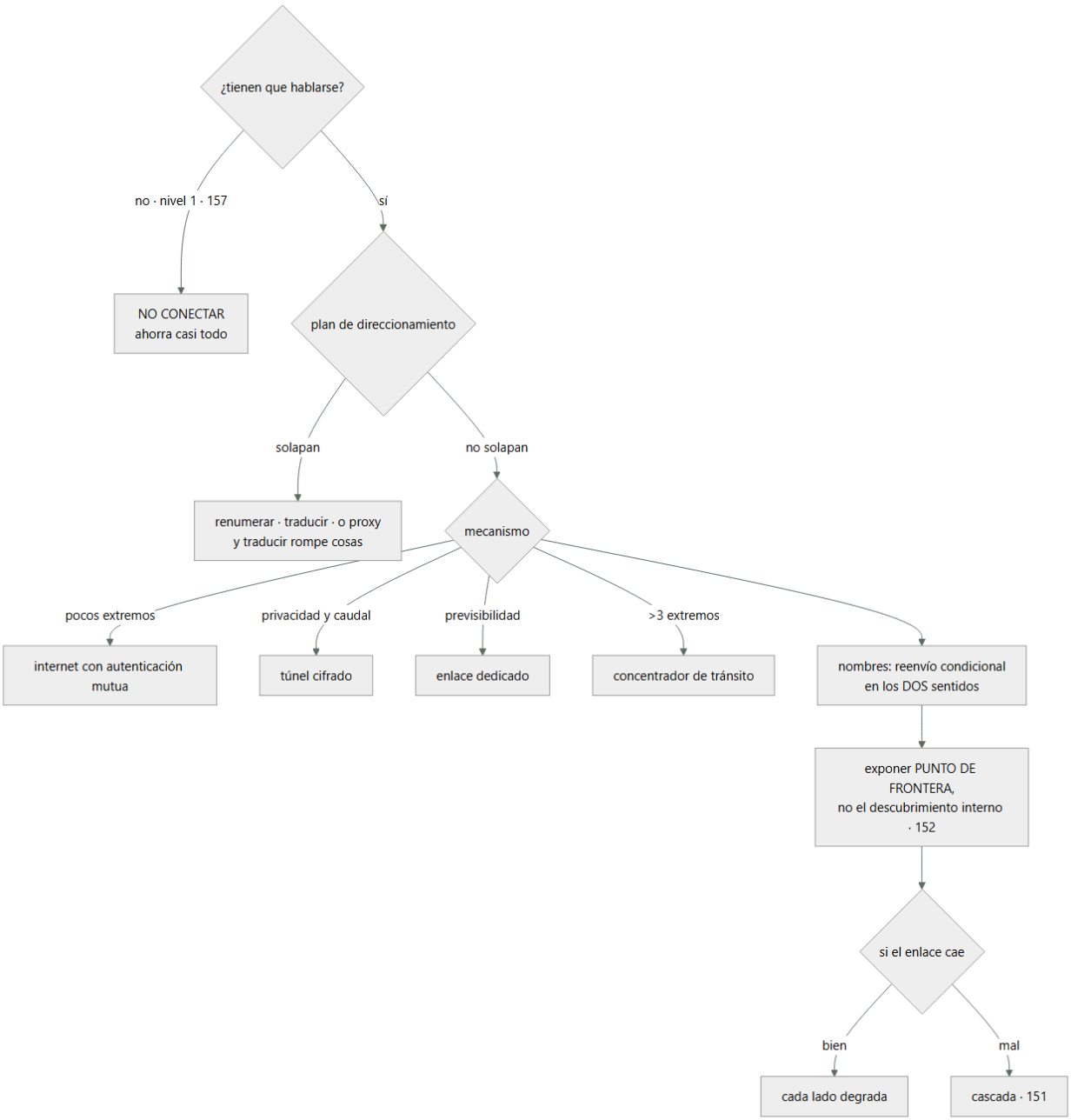
Concepto	Comprensión verificable
nivel de acoplamiento de red	Cuánto necesitan hablarse las cargas de cada proveedor. En el nivel 1 de la clase 157 es cero, y eso ahorra casi todo.
plan de direccionamiento	Reparto de rangos privados entre entornos, regiones y proveedores. Se decide al crear y cambiarlo obliga a renumerar.
solape de rangos	Dos redes con el mismo espacio privado. Impide enrutar entre ellas sin traducir, y traducir rompe cosas.
reenvío condicional	Mandar las consultas de un dominio concreto al resolutor del otro proveedor. Es lo que hace que los nombres privados funcionen entre nubes.
punto de frontera	Extremo estable y documentado por el que se entra a un proveedor, en vez de exponer su descubrimiento interno.
concentrador de tránsito	Punto central por el que pasan las conexiones cuando hay más de dos o tres extremos, para evitar la malla de túneles.

Modelo mental

Multi-cloud es una decisión de negocio y riesgo; duplicar cada componente entre proveedores rara vez es la forma más confiable o económica de lograrla.

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 pregunta previa

Antes de elegir cómo conectar, hay que responder si hace falta:

```
¿qué carga de A necesita hablar con qué carga de B?
¿con qué frecuencia?
¿es síncrono o puede ser asíncrono?              clase 152
¿podría hacerse por una interfaz pública autenticada?
```

Y el resultado sorprende a menudo:

```
nivel 1 de la clase 157 —cargas independientes—
no necesita conectividad privada entre proveedores
→ cada carga vive entera en su proveedor
→ y lo que compartan pasa por interfaces públicas autenticadas
  o por datos que se copian
```

Y el coste de conectar, que es lo que se ahorra al no hacerlo:

```
un plan de direccionamiento común
túneles o enlaces, con su vigilancia y su guardia
```

```
resolución de nombres cruzada
reglas de cortafuegos en los dos lados
salida de datos por el tráfico que cruza          clase 161
un modo de fallo nuevo: el enlace
y un camino de movimiento lateral entre proveedores  clase 133
```

El último merece detenerse: conectar dos nubes crea un camino que antes no existía, y eso cambia el ejercicio de alcance de la clase 133.

Y las tres formas de necesitar menos conectividad, en orden de preferencia:

```
1. QUE NO SE HABLEN
   la carga completa vive en un lado, con sus datos

2. QUE SE HABLEN POR UNA INTERFAZ PÚBLICA AUTENTICADA
   con autenticación mutua y lista de permitidos    clases 135, 136
   → es tráfico por internet, cifrado y acotado
   → suficiente para muchísimos casos, y sin infraestructura nueva

3. QUE SE COMUNIQUEN DE FORMA ASÍNCRONA
   publicando y consumiendo, en vez de llamando       parte 09
   → tolera latencia y cortes
```

Y solo cuando ninguna sirve —volumen alto, latencia baja, requisito de no salir a internet— se monta conectividad privada.

2. El direccionamiento, que sí es el problema

El enlace físico o lógico rara vez falla. Lo que bloquea los proyectos es esto:

```
dos redes privadas con el mismo rango
→ no se pueden enrutar entre sí
```

Y ocurre casi siempre, por tres motivos:

```
los valores por defecto de los proveedores coinciden
cada equipo eligió su rango cuando creó su red
y una adquisición trae rangos que nadie coordinó
```

Las tres salidas, con lo que cuesta cada una:

```
RENUMERAR
  la correcta y la cara
  → hay que cambiar direcciones, reglas y todo lo que las tenga escritas
  → y con recursos gestionados, a veces exige recrearlos

TRADUCIR DIRECCIONES ENTRE LAS DOS
  funciona y rompe cosas:
    lo que lleva direcciones dentro del contenido de los mensajes
    los registros: la dirección que se ve no es la real
    la depuración, que pasa a requerir una tabla de traducción
    y las conexiones iniciadas desde el otro lado, según el caso

PROXY EN LA FRONTERA
  solo para servicios concretos
  → no enruta la red: expone puntos concretos
  → y para pocos servicios es la opción más limpia
```

Y la prevención, que es una decisión de creación y por tanto irreversible —ley 14—:

```
UN PLAN DE DIRECCIONAMIENTO CENTRAL desde el primer día
  un bloque grande reservado para la organización
  repartido por proveedor, región y entorno
  con reserva para crecimiento y para adquisiciones
  y nadie crea una red sin pedir su rango
```

Y el control que lo hace cumplir es el de la clase 139: una política que impide crear una red con un rango no asignado.

Y una advertencia sobre el tamaño, que se equivoca en las dos direcciones:

```
rangos demasiado pequeños   se agotan y hay que añadir bloques sueltos
rangos demasiado grandes    se agota el espacio total de la organización
→ y en entornos con muchas direcciones por instancia —contenedores—
  el consumo es mucho mayor de lo que la intuición sugiere
```

Los mecanismos de enlace, ordenados por coste:

```
INTERNET CON AUTENTICACIÓN MUTUA
  + nada que montar; cifrado de extremo a extremo
  − latencia variable y salida de datos a tarifa normal

TÚNEL CIFRADO SOBRE INTERNET
  + red privada entre los dos
  − caudal limitado, y hay que operar los extremos y su redundancia

ENLACE DEDICADO
  + caudal y latencia previsibles, y tarifa de salida menor
  − caro, con plazo de contratación de semanas

CONCENTRADOR DE TRÁNSITO
  cuando hay más de dos o tres extremos
  → sin él, N extremos exigen del orden de N² túneles
```

Y la última fila es la que decide la arquitectura de red en cuanto hay más de un puñado de redes: la malla de túneles no escala y nadie la mantiene bien.

3. Nombres y descubrimiento

Los nombres privados de cada proveedor solo resuelven dentro de él. Para que una carga de A resuelva un nombre de B hacen falta dos piezas:

```
un resolutor de A que sepa reenviar el dominio de B
un extremo en B que acepte esas consultas desde A
y lo mismo en el sentido contrario, si se necesita
```

Y los tres fallos habituales:

```
RESOLUCIÓN EN UN SOLO SENTIDO
  A resuelve nombres de B y B no resuelve los de A
  → funciona hasta que algo responde con un nombre en vez de una dirección

EL MISMO NOMBRE SIGNIFICA COSAS DISTINTAS
  «base.interno» existe en los dos y apunta a bases distintas
  → un error de configuración manda el tráfico al sitio equivocado,
    sin ningún error
  → la regla: un nombre significa lo mismo en todas partes,
    o se usan sufijos distintos por proveedor

EL RESOLUTOR COMO PUNTO ÚNICO
  si el reenvío cae, deja de resolverse todo lo del otro lado
  → dos extremos, en zonas distintas, y comprobación de salud
```

Y una cuestión que aparece siempre y que conviene decidir de antemano:

```
los tiempos de vida de los registros deciden la velocidad de conmutación
  valores largos       menos consultas, conmutación lenta       clase 109
  valores cortos       conmutación rápida, más carga y más coste
→ y algunos clientes ignoran esos valores y cachean por su cuenta
→ por eso la conmutación por nombre nunca es instantánea
```

El descubrimiento entre proveedores es donde más se complica la gente:

```
mal   extender el descubrimiento interno de un proveedor al otro
      → cada instancia de A conoce cada instancia de B
      → miles de entradas cruzando la frontera, cambiando cada minuto
      → y una malla que hay que operar en los dos lados     clase 152

bien  exponer un PUNTO DE FRONTERA por servicio
      un nombre estable y un extremo con reparto por detrás
      → el otro lado solo conoce ese nombre
      → y dentro de cada proveedor, el descubrimiento sigue siendo interno
```

Y las ventajas del punto de frontera, que son las mismas que las de un contrato:

```
la implementación de cada lado cambia sin afectar al otro
se puede limitar, medir y autorizar en un sitio       clase 118
y el radio de un fallo queda acotado
```

Y una consecuencia de latencia que hay que tener presente al diseñar:

```
latencia entre proveedores, misma zona metropolitana      1-20 ms
entre continentes                                        30-150 ms
```

```
→ una operación conversadora que cruce la frontera 40 veces
  añade segundos                                          clase 152
→ la regla: la frontera se cruza UNA VEZ por operación,
  y a ser posible de forma asíncrona
```

4. Cuando el enlace se cae

Un enlace entre proveedores es un componente más, y falla. Lo que hay que decidir de antemano:

```
¿qué deja de funcionar en A si no ve a B?
¿y en B si no ve a A?
¿alguno de los dos deja de servir a sus usuarios?
```

Y la respuesta correcta, con el vocabulario de la clase 151:

```
cada lado debe DEGRADAR, no caer
→ las dependencias que cruzan la frontera deben ser blandas
→ y si alguna es dura, la disponibilidad de un lado depende del enlace
  y hay que decirlo                                       clase 126
```

Y lo que hay que tener para que eso ocurra:

```
plazos cortos en las llamadas que cruzan                clase 130
cortacircuitos por destino remoto
respuesta alternativa: caché, valor por defecto, encolar
y comprobación de que la dependencia es blanda, ensayada  clase 131
```

Y el ensayo correspondiente, que es de los más informativos de esta parte:

```
cortar el enlace entre proveedores durante 15 minutos
  ¿qué falla en cada lado?
  ¿alguien se entera antes que un cliente?
  ¿se recupera solo al restaurarlo, o hay que intervenir?
```

La tercera pregunta detecta el fallo metaestable de la clase 151: acumulación de reintentos que mantiene el problema después de restaurar el enlace.

Lo que hay que vigilar en la conectividad entre proveedores:

```
disponibilidad y latencia del enlace, medidas de extremo a extremo
caudal usado frente al contratado
pérdida de paquetes
consultas de nombres que fallan por dominio
volumen de datos que cruza, por dirección y por servicio  clase 161
y destinos nuevos que aparecen cruzando la frontera       clase 135
```

La última es de seguridad y de coste a la vez, que es la observación de la clase 144.

Y una decisión que conviene tomar explícitamente:

```
¿el tráfico entre proveedores va cifrado además del túnel?
  sí, siempre: el túnel puede terminar en un sitio que no controlas
  → autenticación mutua de extremo a extremo             clase 136
```

Y la lista de comprobación de la clase:

- ■ está respondida la pregunta de si hace falta conectar
- ■ se ha valorado interfaz pública autenticada y comunicación asíncrona
- ■ existe un plan de direccionamiento central y nadie crea redes sin pedir rango
- ■ no hay solapes; si los hay, está decidido si se renumera o se traduce
- ■ el mecanismo de enlace corresponde al número de extremos y al caudal
- ■ hay concentrador si hay más de tres extremos
- ■ la resolución de nombres funciona en los dos sentidos
- ■ ningún nombre significa cosas distintas en cada proveedor
- ■ los resolutores están duplicados y vigilados
- ■ se expone un punto de frontera por servicio, no el descubrimiento interno
- ■ una operación cruza la frontera una vez, no muchas
- ■ las dependencias que cruzan son blandas y se ha comprobado
- ■ se ha ensayado un corte del enlace y se ha medido la recuperación
- ■ el tráfico va cifrado de extremo a extremo, además del túnel

Y el cierre que enlaza con la clase siguiente: por ese enlace pasan datos, y en cuanto los datos cruzan aparece la partida que la hipótesis de la clase 156 señaló como dominante y que casi nadie estima antes: lo que cuesta sacarlos. Es la materia de la clase 161.

Ejemplo trabajado

CloudShop tiene tres cargas en su segundo proveedor y un proyecto abierto para «conectar las dos nubes». El ejercicio empieza por la pregunta previa y termina conectando una sola cosa.

La pregunta previa, aplicada a las tres cargas.

```
CARGA 1: análisis
  ¿necesita hablar con A?   sí, para leer datos
  ¿con qué frecuencia?      una vez al día, por lotes
  ¿síncrono?                no
  → ASÍNCRONO: A publica los datos al lago del segundo proveedor
  → conectividad privada necesaria: NINGUNA

CARGA 2: datos de tres clientes en su región
  ¿necesita hablar con A?   sí, el flujo de compra los consulta
  ¿con qué frecuencia?      en cada petición de esos clientes
  ¿síncrono?                sí
  → hace falta un camino, y se estudia cuál

CARGA 3: celda de un cliente entero
  ¿necesita hablar con A?   solo para identidad y observabilidad
  → identidad ya federada                              clase 159
  → observabilidad por interfaz pública autenticada     clase 162
  → conectividad privada necesaria: NINGUNA
```

Dos de tres no necesitaban conectividad, y la que la necesitaba se examinó con la segunda pregunta:

```
CARGA 2, opciones
  interfaz pública con autenticación mutua
    latencia añadida                          14 ms
    coste                                     salida a tarifa normal
    montaje                                   nada nuevo
  túnel cifrado
    latencia añadida                          11 ms
    coste                                     420 €/mes + operarlo
  enlace dedicado
    latencia añadida                           6 ms
    coste                                   2.100 €/mes + 6 semanas

volumen que cruzaría                          41 GB/mes
requisito de no salir a internet              ninguno
```

```
decisión   interfaz pública con autenticación mutua
motivo     14 ms frente a 6 ms no cambia ningún escenario de la clase 145,
           y evita montar y operar un enlace
revisar si el volumen supera 1 TB/mes o aparece un requisito de
           no atravesar internet
```

Coste evitado: 2.100 € al mes y seis semanas de proyecto, por hacer la pregunta antes de elegir la tecnología.

El solape de rangos, que apareció después.

La cuenta heredada de la adquisición (clase 157) sí necesitaba conectarse para migrar:

```
rango de la red principal                  10.0.0.0/8, en uso parcial
rango de la cuenta heredada                10.0.0.0/16   ← solapa
rangos de las redes del segundo proveedor  10.10.0.0/16  ← solapa
```

Y las tres opciones, evaluadas:

```
renumerar la cuenta heredada
  recursos afectados                           41
  reglas con direcciones escritas              88
  recursos que había que recrear                6
  estimación                                 3 semanas

traducir direcciones
  estimación                                  4 días
  lo que rompía
    los registros mostraban direcciones traducidas
    dos aplicaciones enviaban su dirección dentro del mensaje
    la depuración exigía consultar una tabla

proxy para los 3 servicios que había que alcanzar
```

```
  estimación                                  2 días
  limitación                  solo esos 3 servicios, sin enrutar la red

decisión   proxy, porque la conexión era temporal: solo para migrar
y después  la cuenta heredada se apagó                    clase 157
```

Y la prevención, para que no vuelva a ocurrir:

```
plan de direccionamiento central             no había         sí
bloque reservado para la organización             —      un rango grande
reparto por proveedor, región y entorno           —          sí
reserva para adquisiciones                        —          sí
política que impide crear redes con rango
no asignado                                      no          sí
solapes detectados al inventariar                 4           0
```

Los cuatro solapes se corrigieron: dos renumerando redes que aún no tenían carga, y dos documentando que esas redes nunca se conectarán entre sí.

Los nombres.

Al conectar la carga 2 por interfaz pública, no hizo falta reenvío condicional. Pero apareció el segundo fallo del apartado tercero:

```
el nombre «pedidos.interno» existía en los dos proveedores
en A apuntaba al servicio real
en B apuntaba a un servicio de pruebas creado durante la integración
→ una configuración copiada de A a B mandó tráfico al de pruebas
→ 40 minutos de pedidos escritos en una base de pruebas
→ recuperados desde el registro de eventos                clase 114

sufijo de dominio por proveedor              el mismo     distinto
un nombre, un significado                       no            sí
inventario de nombres internos               no había      41 nombres
nombres duplicados con significados distintos    3             0
```

El punto de frontera.

La primera propuesta para la carga 2 era extender la malla:

```
extender el descubrimiento interno entre proveedores
  entradas que cruzarían la frontera            ~1.800, cambiando
  malla que operar en los dos lados                    sí
  estimación                                     4 semanas

punto de frontera por servicio
  nombres estables expuestos                           2
  reparto detrás de cada uno                    interno de cada lado
  estimación                                       3 días

entradas cruzando la frontera                ~1.800             2
componentes nuevos que operar                   2               0
límite y medición por consumidor                no          sí, clase 118
radio de un fallo                          todo el otro lado   ese servicio
```

El ensayo del corte.

Aunque no había enlace privado, sí había dependencia entre proveedores para la carga 2. Se ensayó cortando el acceso:

```
corte de 15 minutos, provocado

lo que se esperaba
  el flujo de compra sigue para el 96 % de los clientes
  los 3 clientes de la carga 2 reciben datos cacheados

lo que ocurrió
  el flujo de compra siguió                                  ✓
  los 3 clientes recibieron error, no datos cacheados        ✗
    → la dependencia estaba declarada como blanda y no lo era
  al restaurar, 6 minutos de latencia alta                   ✗
    → reintentos acumulados                          clase 151
  nadie se enteró antes que un cliente                       ✗
    → faltaba alerta sobre el camino entre proveedores

dependencia realmente blanda                    no            sí
```

```
  (caché con valor caducado servible)                    clase 111
plazo en llamadas que cruzan                    30 s           2 s
cortacircuitos por destino remoto                no            sí
alerta sobre el camino entre proveedores         no            sí
tiempo de recuperación tras restaurar          6 min          25 s
ensayo del corte                             no se hacía    trimestral
```

A los seis meses.

```
cargas en el segundo proveedor                    3             3
que necesitan conectividad privada                —             0
enlaces dedicados contratados                  proyecto         0
coste mensual de conectividad                  2.100 € previsto 0 €
plan de direccionamiento central                  no            sí
solapes de rango                                  4             0
nombres con significado duplicado                 3             0
entradas de descubrimiento cruzando frontera   ~1.800 previstas 2
dependencias entre nubes realmente blandas     0 de 1         1 de 1
ensayo de corte                                   no        trimestral
```

La lección que esta clase traslada a la parte 13: el proyecto de «conectar las dos nubes» terminó sin ninguna conectividad privada, porque dos de las tres cargas no necesitaban hablarse y la tercera se resolvió con una interfaz pública autenticada que añadía ocho milisegundos más que un enlace dedicado de dos mil cien euros al mes. Y de los tres problemas que sí aparecieron —solape de rangos, un nombre con dos significados y una dependencia que se creía blanda—, ninguno era del cable: dos eran de direccionamiento y nombres, y el tercero solo se descubrió cortando el enlace a propósito.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-13-multicloud-hybrid-disaster-recovery/160-conectividad-transito-dns-y-service-discovery/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 red-multicloud 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 red-multicloud 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 monta un enlace caro entre proveedores y apenas se usa	No se preguntó si las cargas necesitaban hablarse ni se valoró interfaz pública o comunicación asíncrona	Responde primero qué carga necesita qué, con qué frecuencia y si puede ser asíncrono.
No se puede enrutar entre dos redes	Los rangos privados solapan, por valores por defecto o por una adquisición	Plan de direccionamiento central con política que impida crear redes sin rango asignado; y si ya solapan, renumera o expón proxies para servicios concretos.
El tráfico va a un servicio equivocado sin ningún error	El mismo nombre existe en los dos proveedores con significados distintos	Sufijos de dominio distintos por proveedor e inventario de nombres internos.
Miles de entradas de descubrimiento cruzan la frontera y cambian constantemente	Se extendió el descubrimiento interno de un proveedor al otro	Expón un punto de frontera estable por servicio y deja el descubrimiento interno dentro de cada nube.
Una caída del enlace tumba un lado entero	Las dependencias que cruzan son duras aunque estén declaradas como blandas	Plazos cortos, cortacircuitos, respuesta alternativa y ensayo de corte trimestral.
Tras restaurar el enlace, el sistema tarda en recuperarse	Reintentos acumulados que sostienen el problema	Presupuesto de reintentos y colas acotadas, como en la clase 151.

■ Seguridad, ética y costo

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

■ Preguntas de comprobación

1. ¿Qué pregunta hay que responder antes de elegir cómo conectar dos proveedores?
2. ¿Por qué el direccionamiento es el problema y no el enlace?
3. ¿Qué tres fallos aparecen al resolver nombres entre proveedores?
4. ¿Por qué es un error extender el descubrimiento interno y qué se hace en su lugar?
5. ¿Qué revela un ensayo de corte del enlace que no revela ninguna revisión?

Referencias

- AWS (2025). Hybrid connectivity: VPN, Direct Connect and Transit Gateway — mecanismos y cuándo hace falta un concentrador. <https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/introduction.html>
- Google Cloud (2025). Hybrid and multicloud networking patterns — patrones de conexión y sus compromisos. <https://cloud.google.com/architecture/hybrid-multicloud-network-topologies>
- Azure (2025). DNS resolution across on-premises and cloud — reenvío condicional y extremos de resolución. <https://learn.microsoft.com/azure/architecture/hybrid/hybrid-dns-infra>
- RFC 1918 y RFC 6890 (IETF). Espacios de direcciones privadas — base del plan de direccionamiento. <https://www.rfc-editor.org/rfc/rfc6890.html>
- Google SRE (2025). Load balancing and DNS TTL trade-offs — velocidad de conmutación frente a carga de resolución. <https://sre.google/sre-book/load-balancing-frontend/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 13 en PDF · Manual integral

Evaluación — 160 Conectividad, tránsito, DNS y service discovery

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 red-multicloud 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.

161 — Replicación de datos, soberanía y costos de egress

Parte: 13 — Multi-cloud, híbrido, migración y recuperación Nivel: experto · Horas estimadas: 4 Laboratorio: data · Estado: EXECUTABLECORE

Propósito

Mover datos entre zonas, regiones y proveedores sabiendo lo que cuesta, que casi nunca se estima antes. La clase explica la asimetría que da forma a toda la nube —entrar es gratis y salir se paga—, hace la aritmética de replicar de forma continua frente a mover una vez, y enumera los sitios donde la salida se esconde, empezando por el que suele ser la mayor partida y el que menos se mira: el tráfico entre zonas del mismo proveedor. Y termina con las dos decisiones que se toman al crear y no se deshacen: dónde vive cada dato y en qué sentido se replica.

Resultados de aprendizaje

Al finalizar podrás:

1. Enumerar qué tráfico se factura y en qué dirección.
2. Calcular el coste de replicar de forma continua frente a mover una vez.
3. Encontrar la salida escondida, que suele estar dentro del mismo proveedor.
4. Reducir el movimiento por el orden correcto, empezando por no moverlo.
5. Elegir el sentido de la replicación sabiendo qué implica el doble sentido.

Conceptos centrales

Concepto	Comprensión verificable
asimetría de entrada y salida	Introducir datos suele ser gratuito y extraerlos se factura. Es el motivo estructural de que los datos atraigan al cómputo.
gravedad de los datos	Tendencia del cómputo a acercarse a donde están los datos, porque moverlos cuesta dinero y tiempo.
tráfico entre zonas	Comunicación entre zonas de la misma región. Se factura en varios proveedores y suele ser la mayor partida escondida.
coste de mantener frente a coste de mover	Replicar cuesta el ritmo de cambio por el tiempo; migrar cuesta el tamaño total, una vez.
replicación por cambios	Enviar los hechos o las diferencias en vez del estado completo. Reduce el volumen en uno o dos órdenes de magnitud.
colocación	Decisión de dónde vive cada dato. Se toma al crear y arrastra al cómputo detrás.

Modelo mental

Multi-cloud es una decisión de negocio y riesgo; duplicar cada componente entre proveedores rara vez es la forma más confiable o económica de lograrla.

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

■ Flujo de razonamiento

Desarrollo

1. Qué se factura, y la asimetría

El modelo de coste del movimiento de datos, con la forma que tiene en los tres proveedores:

```
DENTRO DE UNA ZONA                  normalmente gratuito
ENTRE ZONAS DE LA MISMA REGIÓN      se factura, y en algunos casos
                                    en los dos sentidos
ENTRE REGIONES                      se factura, más caro
HACIA INTERNET                      se factura, por tramos de volumen
DESDE INTERNET HACIA DENTRO         normalmente gratuito
POR ENLACE PRIVADO                  tarifa por gigabyte menor,
                                    más un coste fijo del enlace
```

Y la segunda línea es la que produce las sorpresas: el tráfico entre zonas se factura y ocurre dentro del mismo proveedor y de la misma región, así que nadie lo asocia con «salida de datos».

Y la asimetría de las líneas cuarta y quinta explica la forma de la nube:

```
meter datos es gratis; sacarlos cuesta
→ y por eso el cómputo se acerca a los datos y no al revés
→ es lo que se llama gravedad de los datos, y no es una metáfora:
  es una consecuencia del modelo de precios
```

Y tres consecuencias prácticas que conviene tener presentes:

```
empezar a usar un proveedor es barato; dejar de usarlo no
un servicio de análisis en otro proveedor paga la salida CADA VEZ
y una copia de seguridad fuera se paga al crearla y al restaurarla
```

La aritmética de replicar frente a migrar, que se confunde siempre:

```
MIGRAR UNA VEZ
  coste = TAMAÑO TOTAL × tarifa
  118 TB × ~0,08 €/GB ≈ 9.400 €, una vez

REPLICAR DE FORMA CONTINUA
  coste = RITMO DE CAMBIO × tiempo × tarifa
  si cambian 40 GB al día:
  40 GB × 30 días × ~0,08 €/GB ≈ 96 €/mes
```

Y la lectura, que es contraintuitiva:

```
mantener una copia al día suele ser MUCHO más barato que moverla una vez
→ porque los datos cambian mucho menos de lo que ocupan
→ y por eso una réplica continua es viable donde una migración asusta
```

Y el corolario para la continuidad, que la clase 166 desarrollará:

```
la copia continua se paga poco a poco
la RESTAURACIÓN se paga entera, de golpe, y justo cuando hay una crisis
→ y por eso hay que estimarla antes y saber cuánto tarda
```

2. Dónde se esconde la salida

Al desglosar una factura de red, las partidas aparecen en un orden que casi nadie espera:

```
TRÁFICO ENTRE ZONAS DEL MISMO CLÚSTER      suele ser la mayor
  cada llamada entre servicios puede cruzar zonas
  y el reparto por defecto no sabe nada de zonas

TELEMETRÍA ENVIADA FUERA                    clase 132
  registros, métricas y trazas hacia otro proveedor o hacia internet

COPIAS DE SEGURIDAD A OTRO SITIO
  se paga al crearlas y al restaurarlas

DESCARGA DE IMÁGENES DE CONTENEDOR
  cada instancia nueva descarga; con autoescalado, muchas veces

CONSULTAS QUE LEEN DE OTRA REGIÓN
  un motor de análisis que lee datos de otra región paga por cada
  consulta                                          clase 112

RESPUESTAS AL USUARIO
```

```
  la partida legítima y la que casi siempre se supone que es la única

RÉPLICAS DE BASES ENTRE REGIONES
Y EL TRÁFICO ENTRE PROVEEDORES                    clase 160
```

Y las tres primeras suelen sumar más que la sexta, que es la única que la gente tiene en la cabeza.

El tráfico entre zonas merece detalle porque es el más fácil de reducir:

```
causa   el reparto elige instancia sin mirar la zona
        y la base o el caché están en otra zona

efecto  una operación con 4 llamadas internas cruza zonas
        unas 3 veces de media

corrección
  reparto consciente de zona: preferir la instancia local y
  salir de zona solo si no hay sana
  colocar caché y réplicas de lectura en cada zona
  y medir la proporción de tráfico que cruza
```

Y el compromiso que hay que aceptar, que es un traslado de la clase 155:

```
reparto local        menos coste y menos latencia
                     y menos margen: si la zona local se degrada,
                     hay que salir
→ se resuelve con umbral: local mientras haya suficientes instancias sanas
```

Y una advertencia sobre la medición, porque cuesta encontrar el desglose:

```
la factura agrupa el tráfico por tipo, no por servicio
→ hace falta atribuirlo con etiquetas y con registros de flujo
  muestreados                                  clases 135, 142
→ y esa atribución es la que convierte «la red cuesta mucho»
  en «este servicio cuesta mucho»
```

3. Reducir, en el orden correcto

```
1. NO MOVER LOS DATOS: LLEVAR EL CÁLCULO
   ejecutar la consulta donde están y traer el resultado
   → un agregado de 4 KB en vez de 40 GB
   → es la reducción mayor con diferencia, y la menos considerada

2. FILTRAR Y AGREGAR EN EL ORIGEN
   no exportar la tabla entera para quedarse con tres columnas
   → es la lección de la clase 112 aplicada al transporte

3. COMPRIMIR Y USAR FORMATOS COLUMNARES
   factores de 5 a 20 en datos analíticos              clase 112

4. REPARTO Y COLOCACIÓN CONSCIENTES DE ZONA
   lo del apartado anterior

5. CACHÉ EN LA FRONTERA
   servir desde el borde lo que se pide muchas veces      clase 111

6. ENLACE PRIVADO CON TARIFA MENOR
   la última opción: reduce el precio por gigabyte y no el volumen
   → y solo compensa con volumen alto y sostenido
```

Y la observación que ordena la lista: las cinco primeras reducen el volumen; la sexta solo reduce el precio. Empezar por la sexta es lo habitual y lo peor.

La replicación entre proveedores, con sus tres formas:

```
COPIA COMPLETA PERIÓDICA
  simple, y mueve todo cada vez
  → solo para conjuntos pequeños o poco frecuentes

DIFERENCIAS
  mueve lo que cambió desde la última vez
  → mucho menos volumen, y exige llevar la cuenta

POR CAMBIOS O POR HECHOS                          clases 114, 115
  se publica lo que ocurre y el otro lado lo aplica
  → el volumen es el ritmo de cambio
```

```
  → y de regalo, el otro lado puede construir su propia vista
  → es casi siempre la respuesta correcta
```

Y el sentido de la replicación es la decisión estructural:

```
UN SENTIDO       A escribe, B copia
  + no hay conflictos: un solo escritor              ley 21
  + B puede servir lecturas y respaldos
  − B no puede escribir; si A cae, B es de solo lectura
    hasta que alguien decida promoverlo               clase 166

DOS SENTIDOS     los dos escriben
  − conflictos garantizados                          clase 149
  − y la latencia entre nubes hace inviable coordinar
  → solo tiene sentido si los datos se PARTEN de modo que cada
    uno tenga un solo escritor, y entonces no es doble sentido:
    son dos conjuntos independientes
```

Y el retardo, que es mucho peor que dentro de un proveedor:

```
replicación dentro de una región          milisegundos
entre regiones del mismo proveedor        decenas a cientos de ms
entre proveedores                         segundos, y minutos bajo carga
→ y todo lo que dependa de esa copia debe tolerarlo    clase 149
```

4. Dónde vive cada dato

La colocación es una decisión de creación, y arrastra todo lo demás:

```
donde vive el dato, acaba viviendo el cómputo que lo usa
y moverlo después cuesta el tamaño total y un proyecto
```

Y los criterios que la deciden, en orden:

```
1. REQUISITO LEGAL O CONTRACTUAL                       clase 141
   residencia por cliente, por país o por categoría de dato
   → es el único criterio que no se negocia

2. DÓNDE ESTÁ QUIEN LO USA
   latencia para los usuarios y para el cómputo que lo procesa

3. DÓNDE ESTÁ EL RESTO DE LOS DATOS CON LOS QUE SE CRUZA
   partir un conjunto que se consulta junto multiplica el tráfico

4. COSTE
```

Y la revisión de residencia entre proveedores es más difícil que dentro de uno, porque:

```
los mapas de regiones no coinciden: lo que en uno es una región,
  en otro son dos, o no existe
los metadatos y el plano de control se procesan en sitios distintos
  y con reglas distintas                               clase 141
y la copia de seguridad, la telemetría y el soporte de CADA proveedor
  tienen su propia geografía
```

Y el método de la clase 141, ampliado a dos proveedores:

```
para cada categoría de dato y cada cliente
  ¿en qué región de qué proveedor vive el dato principal?
  ¿y sus copias?
  ¿y sus réplicas?
  ¿y sus registros y su telemetría?
  ¿y quién puede acceder desde dónde?
→ y la respuesta se comprueba, no se supone
```

Y la técnica que resuelve el borrado cuando el dato ha viajado, que ya apareció en la clase 141 y aquí es más necesaria:

```
clave de cifrado por cliente y por región
→ destruir la clave hace ilegible el dato allí donde haya llegado
→ incluidas copias en el otro proveedor
```

Y lo que hay que vigilar de forma continua:

```
volumen que sale, por servicio y por destino          clase 142
proporción del tráfico interno que cruza zonas
retardo de replicación entre proveedores, en segundos
coste de salida por unidad de negocio
```

```
y datos que aparecen en una región no autorizada
```

La última es de cumplimiento y se comprueba, no se confía: una réplica creada por comodidad en otra región es un incumplimiento silencioso.

Y la lista de comprobación de la clase:

- está desglosado qué tráfico se factura y en qué dirección
- está medida la proporción de tráfico interno que cruza zonas
- el reparto es consciente de zona, con umbral de salud
- la telemetría no cruza fronteras innecesariamente
- las imágenes se descargan de un registro local a cada región
- se lleva el cálculo al dato antes que el dato al cálculo
- la replicación mueve cambios, no estado completo
- el sentido de la replicación es único, o los datos están partidos
- el retardo entre proveedores está medido y lo toleran quienes dependen
- la colocación de cada categoría de dato está decidida y comprobada
- existe forma de borrar un dato que ha viajado
- el coste de restaurar está estimado, no solo el de replicar

Y el cierre que enlaza con la clase siguiente: con datos y cargas repartidos entre proveedores, hace falta saber qué pasa en los dos a la vez, con las mismas señales y sin duplicar el coste que la parte 10 costó reducir. Es la materia de la clase 162.

Ejemplo trabajado

CloudShop desglosa su factura de red antes de diseñar la replicación entre proveedores. La mayor partida no cruza ninguna frontera: está dentro de la misma región.

El desglose, tras atribuir con etiquetas y registros de flujo muestreados.

```
tráfico entre zonas del mismo clúster              880        31 %
respuestas a usuarios por internet                 610        21 %
telemetría enviada al segundo proveedor            410        14 %
descarga de imágenes de contenedor                 320        11 %
consultas de análisis que leen de otra región      280        10 %
copias de seguridad hacia otro proveedor           190         7 %
réplicas de base entre regiones                    160         6 %
                                                  ■■■■■
                                                2.850 €/mes

coste de cómputo                               18.520 €/mes
proporción de la red sobre el cómputo              15 %
```

Y el dato que la hipótesis de la clase 156 anticipaba: la salida de datos es una partida de dos cifras porcentuales, y la mayor parte no es lo que la gente imagina.

La partida mayor: tráfico entre zonas.

```
instancias repartidas en 3 zonas
reparto                                    sin conciencia de zona
probabilidad de que una llamada cruce zona            2 de 3
llamadas internas por operación                          4
cruces de zona por operación                          ~2,7
```

Y la corrección, que no cambió ninguna línea de código de aplicación:

```
reparto                                  aleatorio     prefiere zona local
umbral de salud para salir de zona          —          si hay < 2 sanas
caché por zona                             no              sí
réplica de lectura por zona                 1              3

cruces de zona por operación               2,7            0,3
coste de tráfico entre zonas            880 €/mes      190 €/mes
latencia p99 interna                      14 ms          9 ms
```

Seiscientos noventa euros al mes y cinco milisegundos, por una opción de configuración del reparto.

Las otras tres partidas escondidas.

```
TELEMETRÍA (410 €/mes)
  se enviaba al segundo proveedor porque allí estaba la herramienta
  corrección   recolector local que agrega y filtra antes de enviar
                                                       clase 124
```

```
  volumen enviado                   de 1,4 TB/mes a 210 GB/mes
  coste                             410 € → 62 €

DESCARGA DE IMÁGENES (320 €/mes)
  las instancias descargaban del registro de otra región
  corrección   réplica del registro en cada región
  coste        320 € → 40 €

CONSULTAS DE ANÁLISIS (280 €/mes)
  el motor leía los ficheros del lago de otra región
  corrección   llevar el cálculo al dato: ejecutar la consulta
               en la región del lago y traer el resultado
  volumen      de 3,1 TB/mes a 8 GB/mes
  coste        280 € → 3 €
```

La última es la regla número uno del apartado tercero, con su factor: de tres teratabytes a ocho gigabytes, porque lo que viajaba era el resultado y no los datos.

El diseño de la replicación al segundo proveedor.

Los tres clientes con requisito de región propia necesitaban sus datos allí.

```
opciones evaluadas
  copia completa diaria
    volumen                         410 GB × 30 = 12,3 TB/mes
    coste                           ~980 €/mes
  diferencias diarias
    volumen                         ~1,2 TB/mes
    coste                           ~96 €/mes
  por hechos, continuo                              clase 115
    volumen                         ~310 GB/mes
    coste                           ~25 €/mes
    retardo                         segundos

decisión   por hechos, continuo
motivo     un orden de magnitud más barato y el retardo es aceptable
           para el escenario del cliente
```

Y el sentido, decidido explícitamente:

```
¿los tres clientes escriben en el segundo proveedor?
  sí, es su región y allí se sirve su tráfico
¿escriben también en el principal?
  no
→ cada cliente tiene UN escritor, en su proveedor           ley 21
→ no es replicación de doble sentido: son conjuntos partidos
→ y lo que se replica hacia el principal es solo lo agregado
  para informes, en un sentido
```

Y el retardo medido, para que quien dependa de él lo sepa:

```
retardo de replicación, mediana                        2,1 s
percentil 99                                           9 s
durante el proceso nocturno                            41 s
dependencias que lo toleran                         3 de 3   ✓
```

El coste de restaurar, estimado antes de necesitarlo.

```
copia continua hacia el segundo proveedor        25 €/mes
volumen acumulado                                 4,1 TB
coste de traerlo de vuelta en una restauración   ~330 €
tiempo estimado de transferencia                 6 h 20
```

Y esa última cifra pasó al plan de continuidad de la clase 166: seis horas y veinte minutos solo de transferencia, que es lo que hay que sumar al objetivo de recuperación.

La residencia, comprobada en los dos proveedores.

```
datos principales de los 3 clientes        —           su región    ✓
copias de seguridad                        —           su región    ✓
réplicas                                   —           su región    ✓
telemetría de esas cargas                region de A  ← INCUMPLÍA
registros de esas cargas                 region de A  ← INCUMPLÍA
soporte del proveedor                    documentado   documentado
metadatos                                fuera, documentado
```

Dos incumplimientos, los dos en la misma categoría que la clase 141 ya había señalado: registros y telemetría.

```
telemetría de esas cargas               región de A    su región
registros                               región de A    su región
coste adicional                              —          +40 €/mes
comprobación automática de residencia         no        semanal
datos en región no autorizada                 2            0
```

A los seis meses.

```
coste mensual de red                    2.850 €          520 €
proporción sobre el cómputo                15 %           2,8 %
tráfico entre zonas                       880 €          190 €
cruces de zona por operación                2,7            0,3
telemetría enviada fuera                  1,4 TB/mes     210 GB/mes
consultas que leen de otra región         3,1 TB/mes       8 GB/mes
replicación entre proveedores            no había        25 €/mes
sentido de la replicación                    —           único
retardo medido y tolerado                    no             sí
coste de restaurar, estimado                 no        330 € y 6 h 20
incumplimientos de residencia                 2              0
```

La lección que esta clase traslada a la parte 13: la mayor partida de la factura de red no cruzaba ninguna frontera: era tráfico entre zonas de la misma región, provocado por un reparto que no sabía nada de zonas, y se redujo en un 78 % cambiando una opción de configuración. Y la replicación entre proveedores, que era lo que motivaba el ejercicio, resultó costar veinticinco euros al mes al mover hechos en vez de estado —frente a los novecientos ochenta de copiar el conjunto entero cada día—, porque los datos cambian mucho menos de lo que ocupan.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-13-multicloud-hybrid-disaster-recovery/161-replicacion-de-datos-soberania-y-costos-de-egress/lab.py
```

El laboratorio selecciona el motor de práctica data 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 estrategia-datos-multicloud 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 modelo de datos ligado a patrones de acceso. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye estrategia-datos-multicloud 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 factura de red es alta y el tráfico hacia internet es pequeño	El grueso es tráfico entre zonas de la misma región, que también se factura	Reparto consciente de zona con umbral de salud, caché y réplicas por zona, y atribución del tráfico por servicio.
Replicar un conjunto grande parece inviable por su coste	Se está calculando el tamaño total en vez del ritmo de cambio	Replica cambios o hechos: el coste es el ritmo de cambio por el tiempo, casi siempre uno o dos órdenes de magnitud menor.
Un motor de análisis en otra región dispara el coste	Se mueven los datos al cálculo en vez del cálculo a los datos	Ejecuta la consulta donde están los datos y trae solo el resultado.
Se contrata un enlace privado y el coste apenas baja	El enlace reduce el precio por gigabyte, no el volumen	Agota primero las cinco medidas que reducen volumen; el enlace es la última opción.
Aparecen conflictos al replicar entre proveedores	Se replica en dos sentidos y el dato tiene dos escritores	Un solo sentido, o parte los datos de modo que cada uno tenga un único escritor.
Los datos cumplen la residencia y los registros no	La telemetría y los registros siguen el camino por defecto del proveedor	Comprueba la residencia de datos, copias, réplicas, registros y telemetría en los dos proveedores, y automatiza la comprobación.

■ 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 tráfico se factura y en qué dirección, y qué consecuencia tiene la asimetría?
2. ¿Por qué mantener una copia al día suele ser más barato que moverla una vez?
3. ¿Cuál suele ser la mayor partida escondida y cómo se reduce?
4. ¿Cuál es el orden correcto para reducir el movimiento de datos?
5. ¿Cuándo tiene sentido replicar en dos sentidos?

Referencias

• AWS (2025). Data transfer pricing and cross-AZ charges — qué se factura y en qué dirección. <https://aws.amazon.com/blogs/architecture/overview-of-data-transfer-costs-for-common-architectures/>

• Google Cloud (2025). Network pricing and egress — tarifas por destino y tramos de volumen. <https://cloud.google.com/vpc/network-pricing>

• Azure (2025). Bandwidth pricing and zone-redundant traffic — coste entre zonas y regiones. <https://azure.microsoft.com/pricing/details/bandwidth/>

• Debezium (2025). Change data capture for cross-system replication — replicar cambios en vez de estado. <https://debezium.io/documentation/reference/stable/>

• FinOps Foundation (2025). Data transfer cost allocation — atribuir el tráfico a servicios y equipos. <https://www.finops.org/framework/capabilities/allocation/>

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

Descargar: Parte 13 en PDF · Manual integral

Evaluación — 161 Replicación de datos, soberanía y costos de 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 estrategia-datos-multicloud 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.

162 — Observabilidad y operación entre proveedores

Parte: 13 — Multi-cloud, híbrido, migración y recuperación Nivel: experto · Horas estimadas: 4 Laboratorio: observability · Estado: EXECUTABLECORE

Propósito

Operar dos proveedores sin duplicar lo que la parte 10 costó reducir. La clase se apoya en un principio que resuelve casi todo el problema —un sitio donde mirar, dos sitios donde recoger— y en su consecuencia económica, que viene de la clase 161: no se envía telemetría en bruto de una nube a otra. Después afronta lo que de verdad complica la operación entre proveedores: que la misma cosa se llama distinto en cada uno, que los relojes no coinciden y que la mitad del equipo no sabe intervenir en la mitad del sistema.

Resultados de aprendizaje

Al finalizar podrás:

1. Separar lo que hay que unificar de lo que debe quedarse local.
2. Normalizar nombres y atributos en el punto de recogida.
3. Construir una cronología de incidente que no dependa de las marcas de tiempo.
4. Repartir la guardia cuando no todo el equipo conoce los dos proveedores.
5. Comprobar desde fuera de los dos, que es lo único que dice la verdad.

Conceptos centrales

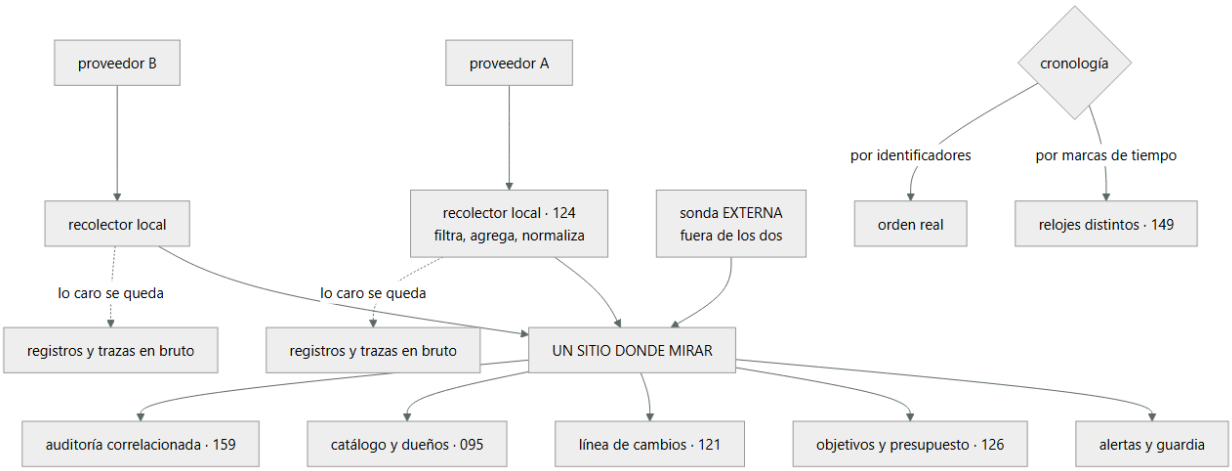
Concepto	Comprensión verificable
un sitio donde mirar	Las decisiones —alertas, objetivos, incidentes— se ven en un único lugar, aunque los datos se recojan en cada proveedor.
normalización en la recogida	Traducir nombres y atributos de cada proveedor a un vocabulario común antes de enviarlos, no al consultarlos.
cronología por identificadores	Reconstruir un incidente encadenando trazas y eventos, no ordenando por marcas de tiempo de relojes distintos.
comprobación externa	Sonda que se ejecuta fuera de los dos proveedores y dice si el usuario puede usar el sistema.
formación cruzada	Proporción del equipo capaz de intervenir en cada proveedor. Es un requisito de disponibilidad, no de carrera profesional.
duplicación de herramientas	Pagar dos veces por lo mismo en cada proveedor. Es la forma que toma aquí el coste de la parte 10.

Modelo mental

Multi-cloud es una decisión de negocio y riesgo; duplicar cada componente entre proveedores rara vez es la forma más confiable o económica de lograrla.

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. Un sitio donde mirar, dos donde recoger

La respuesta ingenua al segundo proveedor es duplicarlo todo:

```
dos consolas, dos sistemas de alertas, dos paneles, dos guardias
→ el coste que la parte 10 redujo se dobla
→ y las alertas de uno se ven en un sitio que nadie mira
```

Y el principio que lo evita:

```
SE UNIFICAN LAS DECISIONES, NO LOS DATOS

un sitio donde mirar      alertas, objetivos, incidentes, cambios
dos sitios donde recoger  registros y trazas se quedan donde nacen
```

Y la razón económica es la clase 161: enviar telemetría en bruto de una nube a otra se paga por gigabyte, y la telemetría es voluminosa por naturaleza.

El reparto concreto:

```
SE UNIFICA
  alertas y encaminamiento de guardia
  objetivos, indicadores y presupuesto de error       clase 126
  la línea de cambios de los dos proveedores          clase 121
  el catálogo de servicios y sus dueños               clase 095
  el registro de auditoría, correlacionado            clase 159
  el panel «¿está funcionando?» por servicio          clase 125
  y el canal y el documento de incidente              clase 127

SE QUEDA LOCAL
  registros en bruto                                  clase 122
  trazas completas                                    clase 124
  métricas de alta resolución
  y todo lo que tenga alta cardinalidad               clase 123
```

Y lo que se envía fuera, que es poco y barato:

```
métricas agregadas de los indicadores del objetivo
el estado de las alertas
los eventos de cambio
un resumen por servicio
y los enlaces para poder saltar al detalle local cuando haga falta
```

Y el último punto es el que hace viable el modelo: desde el sitio unificado se salta al detalle en el proveedor correspondiente, en vez de traer el detalle.

Y una regla de coste que conviene fijar de antemano:

```
el coste de observar el segundo proveedor no debe superar el 20 %
del coste de observar el primero
→ si lo supera, se está enviando en bruto algo que debería agregarse
```

2. Que la misma cosa se llame igual

El obstáculo práctico más constante: cada proveedor nombra las cosas a su manera.

```
el identificador de una instancia
el nombre de la región y de la zona
la clave de la etiqueta de dueño
el nombre de la métrica de uso de procesador
el campo del registro que contiene la identidad
el formato de la marca de tiempo
y el modelo de recursos entero                          clase 159
```

Y si no se normaliza, ocurre lo previsible:

```
un panel por proveedor, con métricas que no se pueden comparar
una alerta por proveedor, con umbrales distintos
y ninguna consulta que abarque los dos
```

La corrección es normalizar en la recogida, no en la consulta:

```
el recolector de cada proveedor traduce a un vocabulario común
  servicio, entorno, versión, región, zona, equipo, inquilino
y añade de dónde viene: proveedor, cuenta o proyecto
```

Y el motivo de hacerlo al recoger y no al consultar:

```
al consultar    hay que traducir en cada panel, cada alerta y cada
                consulta nueva, para siempre
al recoger      se hace una vez, en un sitio, y todo lo demás
                ve un solo vocabulario
```

Y conviene apoyarse en convenciones que ya existen en vez de inventar un vocabulario propio, que es lo que la clase 158 desaconsejaba para los proveedores y vale igual aquí.

Y dos precauciones:

```
lo que no encaje en el vocabulario común se conserva con su nombre
  original, con prefijo de proveedor
  → no se descarta: se marca como específico

y la traducción se versiona y se prueba
  → un cambio en la traducción cambia todos los paneles a la vez
```

Y el catálogo es lo que cierra el círculo:

```
cada servicio del catálogo dice en qué proveedor y qué cuenta vive
→ una alerta puede decir «servicio pedidos, proveedor B, equipo X»
→ y el encaminamiento de guardia funciona sin que nadie lo mantenga
  a mano                                                clase 095
```

3. Los relojes y la cronología

Reconstruir un incidente que atraviesa dos proveedores tiene un problema propio:

```
cada proveedor sincroniza sus relojes con su propia fuente
desviación entre proveedores                        decenas de ms
cuando algo falla                                   segundos
```

Y ordenar una cronología por marcas de tiempo produce secuencias falsas:

```
«el error de B ocurrió 200 ms ANTES de la petición de A que lo causó»
→ y alguien concluye que B falló solo
```

La corrección es la de la clase 124, aplicada aquí:

```
la cronología se construye por IDENTIFICADORES, no por tiempo
  la traza enlaza la petición de A con el trabajo de B
  el evento lleva su identificador de correlación     clase 115
  y el orden lo da la relación de causa, no el reloj

y las marcas de tiempo se usan para medir duraciones DENTRO
de un mismo proveedor, no para ordenar entre ellos
```

Y dos medidas que conviene tener, porque el desajuste también rompe otras cosas:

```
desviación de reloj medida entre proveedores, con alerta
y una prueba periódica: desajustar a propósito y ver qué falla
                                                    clases 131, 149
```

Y lo que suele romperse con desviación entre nubes:

```
caducidad de testigos emitidos en uno y validados en otro   clase 159
firmas con ventana de validez
ordenación de eventos entre proveedores                     clase 114
y ventanas de agregación que no coinciden
```

El incidente que atraviesa los dos necesita además una decisión organizativa:

```
un solo canal y un solo documento, aunque haya dos proveedores
un solo mando, que no tiene por qué conocer los dos    clase 127
y en el canal, alguien capaz de intervenir en cada uno
```

Y la última línea es la que hay que garantizar antes de que ocurra: si a las tres de la madrugada solo hay alguien que conoce el proveedor A, la mitad del sistema no tiene respuesta.

4. Guardia, coste y la comprobación externa

La guardia es donde la clase 157 encontró la cifra que decidió el nivel:

```
personas capaces de operar el proveedor principal              9
personas capaces de operar el segundo                          2
```

Y las tres respuestas, en orden de eficacia:

```
1. AUTOMATIZAR LO ESPECÍFICO DE CADA PROVEEDOR
   que el procedimiento sea el mismo aunque por debajo cambie
   → «reiniciar el servicio X» es un comando, no una consola
                                                  clase 128

2. PROCEDIMIENTOS PARA QUIEN TIENE MENOS CONTEXTO
   con comandos exactos y sin suponer familiaridad
   → y probados por alguien que no conoce ese proveedor

3. FORMACIÓN CRUZADA, MEDIDA
   no como plan de carrera, sino como requisito de disponibilidad
   → objetivo explícito: al menos N personas por proveedor
   → y se comprueba en los ensayos de la clase 131
```

Y una regla que evita el peor caso:

```
ninguna intervención crítica debe requerir conocer un proveedor concreto
→ si la requiere, hay que automatizarla o documentarla hasta que no
→ y mientras tanto, esa persona está de guardia de hecho aunque
  no lo esté en el calendario
```

El coste, con las trampas propias de operar dos nubes:

```
enviar registros o trazas en bruto de una a otra       clase 161
dos licencias del mismo tipo de herramienta
paneles duplicados que hay que mantener dos veces
retención duplicada de lo mismo
y sondas ejecutándose desde los dos proveedores contra los dos
```

Y la medida que lo controla es la de la clase 132, ahora con dos sumandos:

```
coste de telemetría frente a coste de cómputo, por proveedor
y proporción de lo emitido que no consulta nadie, por proveedor
```

La comprobación externa, que es la pieza que solo tiene sentido con varios proveedores:

```
si el sistema vive en A y B, la comprobación que dice
«¿puede el usuario usar esto?» no debe ejecutarse ni en A ni en B
→ porque si el proveedor que la ejecuta se cae, deja de haber señal
  justo cuando hace falta                              ley 13
```

Y lo que debe hacer:

```
ejecutar el recorrido completo del usuario, no un ping
desde varias ubicaciones
con una frecuencia que permita detectar en minutos
y alertar por su propia ausencia de datos             clase 123
```

Y una comprobación adicional que casi nadie tiene y que resuelve discusiones:

```
una sonda que compruebe la ruta ENTRE proveedores    clase 160
→ para distinguir «B está caído» de «no llegamos a B»
```

Y la lista de comprobación de la clase:

- las decisiones están unificadas y los datos en bruto son locales
- no se envía telemetría en bruto entre proveedores
- el coste de observar el segundo no supera una fracción del primero
- los nombres y atributos se normalizan en la recogida
- lo específico se conserva con prefijo, no se descarta
- el catálogo dice en qué proveedor vive cada servicio
- la cronología de incidentes se construye por identificadores
- se mide la desviación de reloj entre proveedores
- un solo canal, un solo mando y alguien capaz de intervenir en cada uno
- lo específico de cada proveedor está automatizado o documentado
- hay objetivo medido de personas capaces por proveedor
- existe sonda externa a los dos, con alerta por ausencia de datos
- existe sonda de la ruta entre proveedores

Y el cierre que enlaza con la clase siguiente: para que la operación sea la misma en los dos, lo que se despliega tiene que declararse de forma comparable, y ahí aparece la capa que la clase 158 marcó como no portable. Cómo se organiza la infraestructura declarada con varios proveedores, y por qué el error es intentar un módulo común, es la materia de la clase 163.

Ejemplo trabajado

CloudShop opera tres cargas en su segundo proveedor. La operación se montó allí copiando lo que había en el principal, y el resultado se descubre durante un incidente: una alerta llevaba once minutos sonando en una consola que nadie miraba.

El incidente que lo puso en evidencia.

```
02:14  la carga de los tres clientes del proveedor B empieza a fallar
02:14  se dispara una alerta en la consola nativa de B
02:25  un cliente escribe a soporte
02:31  soporte localiza a alguien de guardia
02:38  quien está de guardia no tiene acceso operativo a B
02:52  se localiza a una de las dos personas que sí
03:10  mitigado

duración                                          56 min
alerta disparada a los                             0 min
tiempo hasta que alguien la vio                   11 min → nunca:
                                                  se supo por el cliente
```

Y el diagnóstico, con el vocabulario de esta clase:

```
las alertas de B iban a la consola de B, no al sistema unificado
el encaminamiento de guardia no conocía las cargas de B
solo 2 de 9 personas podían intervenir
y el procedimiento suponía familiaridad con ese proveedor
```

La unificación, y lo que costó.

La primera propuesta fue enviar toda la telemetría de B al sistema del proveedor A:

```
volumen de telemetría de las 3 cargas de B        1,1 TB/mes
coste de salida                                     ~88 €/mes
coste de ingesta y retención en A                  ~640 €/mes
total                                               ~728 €/mes
coste de observar el proveedor A                   1.110 €/mes
proporción                                             66 %
```

Sesenta y seis por ciento del coste del principal para tres cargas de quince. Se aplicó el reparto del apartado primero:

```
                                                  datos locales
volumen enviado                    1,1 TB/mes       14 GB/mes
coste de salida                       88 €            1 €
coste de ingesta y retención         640 €           38 €
coste local en B                       0 €          180 €
total                                728 €          219 €
proporción sobre el principal        66 %            20 %
```

Y lo que se envía son las cuatro cosas del apartado primero:

```
indicadores del objetivo, agregados cada 30 s
estado de las alertas
eventos de cambio                                 clase 121
resumen por servicio
```

```
y enlaces para saltar al detalle en B
```

Lo que se unificó, y el efecto en el mismo incidente reproducido.

```
alertas en un solo sitio                        no            sí
encaminamiento de guardia conoce B              no            sí
catálogo con proveedor y cuenta por servicio    no            sí
panel «¿está funcionando?» por servicio      dos formatos    uno
línea de cambios de los dos proveedores         no            sí

ensayo del mismo fallo
  tiempo hasta que alguien lo sabe            56 min         70 s
  tiempo hasta mitigar                        56 min        11 min
```

La normalización, y lo que apareció.

Al traducir a un vocabulario común en los recolectores:

```
atributos que significaban lo mismo con nombres distintos       23
métricas equivalentes con nombres distintos                     31
etiquetas de dueño con claves distintas                          2
formatos de marca de tiempo distintos                            2
```

Y una consecuencia inesperada:

```
al poder comparar, se descubrió que la latencia p99 de la misma
carga era 2,4× mayor en B
causa    el reparto de B no era consciente de zona    clase 161
corrección  la misma que en A, aplicada allí
latencia p99 en B                       de 410 ms a 180 ms
```

No se podía comparar y por eso nadie lo había visto.

Los relojes.

Durante la revisión del incidente, la cronología no cuadraba:

```
según las marcas de tiempo
  02:14:03  error en B
  02:14:03,4  petición en A que lo provocó
→ el error parecía anterior a su causa

desviación medida entre los dos proveedores            310 ms
```

```
cronología construida por               marcas de tiempo   identificadores
desviación de reloj medida                     no        sí, alerta > 100 ms
comportamientos rotos por desajuste,
encontrados en el ensayo                       —              2
  → caducidad de un testigo emitido en A y validado en B
  → una ventana de agregación que no coincidía
```

La guardia.

```
personas capaces de operar A                    9              9
personas capaces de operar B                    2              6
objetivo declarado                         no había        ≥ 5 por proveedor
intervenciones que exigen conocer B         todas        2 de 14
  → las 12 restantes se automatizaron o se documentaron
procedimientos de B probados por alguien
que no lo conocía                          0 de 5        5 de 5
preguntas durante esas pruebas                 —             41
```

Y las cuarenta y una preguntas son la misma medida de la clase 128: cada una era un defecto del procedimiento, y la mayoría fueron permisos y nombres de consola.

La comprobación externa.

```
sondas antes
  desde A hacia A                               sí
  desde A hacia B                               sí
  desde fuera de los dos                        no
```

Y el problema de ese montaje se vio en un ensayo:

```
se simuló una degradación del proveedor A
→ las sondas dejaron de ejecutarse
→ y el sistema de alertas, que también estaba en A, no avisó
→ silencio total durante la caída simulada          ley 13
```

```
sonda externa a los dos                          no             sí
recorrido que ejecuta                         un ping     compra completa
ubicaciones                                      1              4
alerta por ausencia de datos de la sonda         no             sí
sonda de la ruta entre proveedores               no             sí
tiempo de detección en el ensayo repetido    no se detecta     90 s
```

Y la última fila del cuadro anterior resolvió además una discusión recurrente: distinguir «B está caído» de «no llegamos a B», que en dos incidentes anteriores había costado veinte minutos cada vez.

A los cinco meses.

```
alertas en un solo sitio                         no             sí
coste de observar el segundo proveedor      728 €/mes      219 €/mes
proporción sobre el principal                  66 %           20 %
telemetría enviada entre nubes             1,1 TB/mes      14 GB/mes
atributos normalizados                           no         23 + 31
latencia p99 en B                             410 ms          180 ms
cronología por identificadores                   no             sí
desviación de reloj vigilada                     no             sí
personas capaces de operar B                      2              6
intervenciones que exigen conocer B            todas        2 de 14
sonda externa a los dos                          no             sí
tiempo hasta saber de un fallo en B           56 min           70 s
```

La lección que esta clase traslada a la parte 13: el incidente de cincuenta y seis minutos no lo causó ningún fallo del segundo proveedor: lo causó que su alerta sonaba en una consola que nadie miraba y que solo dos personas de nueve podían intervenir. Y al normalizar los nombres para poder comparar apareció, de regalo, que la misma carga era dos veces y media más lenta en B por el mismo problema de reparto que ya se había corregido en A: no se podía comparar, y por eso llevaba meses sin verse.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-13-multicloud-hybrid-disaster-recovery/162-observabilidad-y-operacion-entre-proveedores/lab.py
```

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

Evidencia esperada

El artefacto principal es telemetría correlacionada con una pregunta operativa. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye telemetria-multicloud 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
Una alerta suena en la consola del segundo proveedor y nadie la ve	Se duplicó la operación en vez de unificar las decisiones	Un solo sitio donde mirar: alertas, objetivos, incidentes y cambios unificados; los datos en bruto se quedan donde nacen.
Observar el segundo proveedor cuesta casi tanto como el primero	Se envía telemetría en bruto entre nubes y se paga salida, ingesta y retención	Agrega y filtra en un recolector local y envía solo indicadores, estado de alertas, cambios y enlaces.
No se pueden comparar las métricas de los dos proveedores	Cada uno nombra las cosas a su manera y no se normaliza	Traduce a un vocabulario común en la recogida, conserva lo específico con prefijo y versiona la traducción.
La cronología de un incidente sitúa el efecto antes que la causa	Se ordenó por marcas de tiempo de relojes distintos	Construye la cronología por identificadores de traza y de evento, y vigila la desviación de reloj.
De madrugada nadie puede intervenir en la mitad del sistema	Solo unas pocas personas conocen el segundo proveedor	Automatiza o documenta lo específico, prueba los procedimientos con quien no lo conoce y fija un objetivo medido de personas capaces por proveedor.
Durante una caída del proveedor principal no llega ninguna alerta	Las sondas y el sistema de alertas viven en el proveedor que se cayó	Sonda externa a los dos que ejecute el recorrido del usuario, con alerta por ausencia de datos, más una sonda de la ruta entre proveedores.

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 se unifica y qué se queda local, y por qué?
2. ¿Por qué se normalizan los nombres en la recogida y no en la consulta?
3. ¿Por qué no se puede ordenar una cronología entre proveedores por marcas de tiempo?
4. ¿Qué tres medidas permiten que la guardia funcione con dos proveedores?
5. ¿Por qué la comprobación que dice si el sistema funciona no debe ejecutarse en ninguno de los dos?

Referencias

- OpenTelemetry (2025). Semantic conventions and resource attributes — vocabulario común y normalización. <https://opentelemetry.io/docs/specs/semconv/>
- OpenTelemetry (2025). Collector deployment patterns — recoger local, agregar y exportar a un destino común. <https://opentelemetry.io/docs/collector/deployment/>
- Google SRE (2025). Monitoring across environments and synthetic probes — comprobación externa y ausencia de datos. <https://sre.google/workbook/monitoring/>
- PagerDuty (2025). Incident response with distributed teams — un canal, un mando y encaminamiento por catálogo. <https://response.pagerduty.com/>

• CNCF (2025). Observability across clusters and clouds — patrones de agregación y federación de métricas. <https://www.cncf.io/reports/>

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

Descargar: Parte 13 en PDF · Manual integral

Anterior	Índice	Siguiente
← 161 · Replicación de datos, soberanía y costos de egress	Parte 13 · Programa	163 · Terraform multi-provider y separación de estados →

Evaluación — 162 Observabilidad y operación entre proveedores

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 telemetria-multicloud 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.

163 — Terraform multi-provider y separación de estados

Parte: 13 — Multi-cloud, híbrido, migración y recuperación Nivel: experto · Horas estimadas: 4 Laboratorio: iac · Estado: EXECUTABLECORE

Propósito

Declarar infraestructura en varios proveedores sin cometer el error que la clase 158 ya describió en su forma general: intentar un módulo común que sirva para todos. La clase muestra por qué eso produce un condicional gigante que expone la intersección, propone la alternativa —un módulo por proveedor con la misma interfaz— y dedica el resto a lo que de verdad decide si esto se puede operar: cómo se reparte el estado, que es lo que fija el radio de daño de cada cambio y qué se puede aplicar cuando un proveedor no responde.

Resultados de aprendizaje

Al finalizar podrás:

1. Rechazar el módulo común y sustituirlo por uno por proveedor con la misma interfaz.
2. Repartir el estado por ciclo de vida, radio de daño, proveedor y dueño.
3. Pasar datos entre estados sin acoplarlos de forma rígida.
4. Configurar credenciales por proveedor con identidad federada y acotada.
5. Detectar deriva y ordenar dependencias que cruzan proveedores.

Conceptos centrales

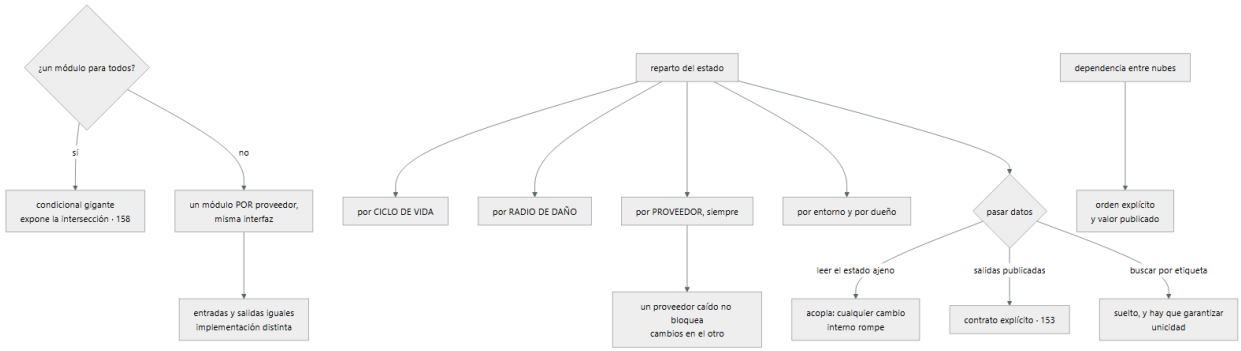
Concepto	Comprensión verificable
módulo por proveedor	Implementación específica con la misma interfaz de entradas y salidas que su equivalente en otro proveedor.
estado	Registro de lo que la herramienta cree haber creado. Su alcance decide el radio de daño y quién puede aplicar cambios.
radio de daño del estado	Todo lo que un solo aplicado erróneo puede destruir. Es el criterio principal para repartir.
salidas publicadas	Contrato explícito de lo que un estado expone a los demás. Sustituye a leer el estado ajeno entero.
proveedor con alias	Varias configuraciones del mismo o de distintos proveedores en un mismo componente, cada una con su identidad.
dependencia entre proveedores	Un recurso de un proveedor que necesita un valor creado en otro. Se resuelve con orden explícito y datos publicados, no con un estado común.

Modelo mental

Multi-cloud es una decisión de negocio y riesgo; duplicar cada componente entre proveedores rara vez es la forma más confiable o económica de lograrla.

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. Un módulo por proveedor, no uno para todos

La propuesta aparece siempre: un módulo «base de datos» que sirva para los dos proveedores, con una variable que diga cuál.

Y falla por los mismos motivos que la capa de abstracción de la clase 158, en versión declarativa:

```
los recursos no tienen la misma forma
  distintos campos obligatorios
  distinta forma de expresar red, copias, cifrado y permisos
  distintos valores por defecto y distintas unidades

→ el módulo se llena de condicionales
→ y solo puede exponer lo que existe en los dos
→ es decir, el mínimo común denominador                clase 157
```

Y la alternativa, que es la de siempre en este programa:

```
UN MÓDULO POR PROVEEDOR, CON LA MISMA INTERFAZ

entradas iguales    nombre, tamaño, entorno, retención, etiquetas
salidas iguales     punto de acceso, identificador, nombre del secreto
implementación      distinta, y usando a fondo lo que ofrece cada uno
```

Y lo que se gana:

```
quien lo usa escribe lo mismo en los dos casos
cada módulo aprovecha lo mejor de su proveedor
no hay condicionales
y se pueden probar por separado                        clase 090
```

Y la disciplina que lo sostiene:

```
las interfaces se documentan y se versionan como contratos  clase 153
y hay una prueba que comprueba que las dos ofrecen las mismas
  entradas y salidas
→ sin esa prueba, divergen en tres meses
```

Y la excepción razonable, para no caer en el extremo contrario:

```
lo que sí puede ser común
  la convención de nombres y etiquetas
  la validación de entradas
  el cálculo de valores derivados
→ eso vive en un módulo auxiliar sin recursos, y lo usan los dos
```

Y una nota sobre las versiones de los complementos de proveedor, que es la clase 138 aplicada aquí:

```
se fijan con huella, como cualquier dependencia
y se actualizan con cadencia, no cuando algo falla
→ una actualización de complemento puede cambiar el plan sin que
  nadie haya tocado nada
```

2. Repartir el estado

El estado es el registro de lo que la herramienta cree haber creado, y su alcance decide dos cosas críticas:

```
el RADIO DE DAÑO de un aplicado erróneo
y qué se puede cambiar cuando algo no responde
```

Los cuatro criterios de reparto, por orden de importancia:

```
1. POR PROVEEDOR, SIEMPRE
   un estado que abarca dos proveedores significa que
   una incidencia en uno bloquea los cambios en el otro
   → y el aplicado falla a medias, dejando lo aplicado sin registrar

2. POR RADIO DE DAÑO
   ¿qué estoy dispuesto a romper de una vez?
   → la red base y la identidad, en estados propios y muy revisados
   → una carga concreta, en el suyo

3. POR CICLO DE VIDA
   lo que cambia junto, junto; lo que cambia a ritmos distintos, aparte
   → la red cambia dos veces al año; el servicio, a diario

4. POR DUEÑO
   cada estado tiene un equipo que lo aplica                clase 095
```

Y el reparto típico que sale de aplicarlos:

```
por proveedor y entorno
  cimientos      organización, cuentas, identidad, políticas
  red            rangos, subredes, salida, resolución de nombres
  datos          bases, almacenes, claves
  plataforma     clúster, registro, herramientas comunes
  por servicio   una por carga
```

Y el compromiso que hay que aceptar, que es un traslado de la clase 155:

```
más estados      menos radio de daño, más coordinación entre ellos
menos estados    más atomicidad, y un error afecta a más
```

Y dos señales de que el reparto está mal:

```
un aplicado rutinario tarda más de unos minutos en planificar
  → el estado abarca demasiado

un cambio pequeño exige aplicar cuatro estados en orden
  → el reparto no sigue el ciclo de vida
```

Lo que debe estar en un estado aparte y muy protegido, con la disciplina de la clase 144:

```
lo que no se puede recrear: bases con datos, claves        clase 136
lo que romperlo deja sin acceso: identidad y red
y lo que borrarlo es irreversible
→ con protección contra destrucción y aprobación de dos personas
```

Y el almacenamiento del propio estado, que es un recurso crítico:

```
remoto, versionado y con bloqueo                          clase 087
cifrado, y con acceso restringido: contiene datos sensibles
y en el proveedor al que gobierna, para no depender del otro
```

La última línea importa: guardar el estado del proveedor B en el proveedor A significa que una caída de A impide arreglar B.

3. Pasar datos entre estados

Con el estado repartido, un estado necesita valores creados por otro. Tres formas, de más acoplada a menos:

```
LEER EL ESTADO AJENO DIRECTAMENTE
  + simple
  − acopla: cualquier reorganización interna del otro rompe
  − y exige permiso de lectura sobre un fichero con datos sensibles

SALIDAS PUBLICADAS COMO CONTRATO
  el estado productor declara qué expone, con nombres estables
  el consumidor lee solo eso
  → es la clase 153 aplicada a la infraestructura
  → y permite cambiar el interior sin romper a nadie

BUSCAR POR ETIQUETA O POR NOMBRE
  el consumidor localiza el recurso por su etiqueta de servicio
  + sin acoplamiento a ningún estado
  − hay que garantizar unicidad y que la etiqueta no cambie
  − y falla en el momento de aplicar, no en el de planificar
```

Y la recomendación práctica:

```
dentro de un mismo dueño y entorno      salidas publicadas
entre equipos distintos                 salidas publicadas, versionadas
entre proveedores                       valores explícitos, pasados
                                        como variables
```

Y la última merece detalle, porque es el caso propio de esta clase:

```
un registro de nombres en A que apunta a un balanceador de B

mal   un estado que abarque los dos, para poder referenciarlo
bien  el estado de B publica la dirección
      un paso de la canalización la toma y la pasa como variable
      al estado de A
      → dos aplicados, en orden, con el valor viajando entre ellos
```

Y la regla que evita el enredo:

```
las dependencias entre estados forman un grafo SIN CICLOS,
y está dibujado
→ si A necesita algo de B y B algo de A, hay que romper el ciclo
  con un valor fijado de antemano —un nombre, un rango— en vez
  de uno generado
```

Las credenciales, que con varios proveedores tienen su propia trampa:

```
cada bloque de proveedor lleva su identidad
y esa identidad viene de la federación de la canalización   clases 098, 159

lo que hay que comprobar
  ¿la identidad de la canalización está acotada por rama y entorno?
  ¿el estado de producción solo se puede aplicar desde el flujo
    de producción?
  ¿alguien puede aplicar a mano con credenciales personales?
```

La última debería ser no, y con el bucle de reconciliación de la clase 103 lo es por construcción.

4. Deriva, orden y pruebas

La deriva de la clase 090 se multiplica con varios estados y proveedores:

```
planificar de forma programada cada estado, sin aplicar
y alertar por diferencias, con la excepción de los campos que
  gestionan otros controladores                        clase 103
```

Y lo que cambia con dos proveedores:

```
las diferencias no son comparables entre sí: cada complemento
  las expresa a su manera
→ conviene normalizar el resultado a «cuántos recursos difieren
  y de qué tipo», que sí se puede comparar y vigilar    clase 162
```

Y la señal que la ley 13 exige, y que aquí se duplica:

```
alerta por ANTIGÜEDAD de la última planificación correcta,
por estado y por proveedor
→ un estado que dejó de planificarse no da ningún error
```

El orden de aplicación entre estados hay que declararlo:

```
cimientos → red → datos → plataforma → servicios
```

Y con dos proveedores, el orden cruza:

```
cimientos de A → cimientos de B → red de A → red de B → …
```

Y la forma de gestionarlo sin construir un orquestador propio:

```
la canalización aplica en orden, por etapas               clase 099
cada etapa publica sus salidas
y un fallo en una etapa detiene las siguientes
→ y hay que poder reanudar desde donde falló, no desde el principio
```

Las pruebas, que son las de la clase 090 con una adición:

```
análisis del plan antes de aplicar                        clase 091
  → ningún borrado de recurso con datos, ninguna región no autorizada
comprobación de que los módulos equivalentes tienen la misma interfaz
entorno efímero que crea y destruye de verdad             clase 104
```

```
  → en los dos proveedores, no solo en el principal
y prueba de portabilidad: que el módulo del segundo proveedor
  se aplica limpio                                          clase 158
```

Y una comprobación que evita sorpresas caras:

```
en el plan, avisar cuando un cambio implique DESTRUIR Y RECREAR
→ es la ley 14 hecha visible: hay campos que no se pueden cambiar
  en sitio, y el plan lo dice si alguien lo lee
→ y con datos dentro, eso es una pérdida
```

Y la lista de comprobación de la clase:

- no existe un módulo común para varios proveedores
- los módulos equivalentes tienen la misma interfaz, comprobada
- las versiones de los complementos están fijadas con huella
- el estado está repartido por proveedor, radio de daño, ciclo de vida y dueño
- ningún estado abarca dos proveedores
- el estado de cada proveedor se guarda en ese proveedor
- el estado está cifrado, versionado, con bloqueo y acceso restringido
- los datos entre estados pasan por salidas publicadas o variables
- el grafo de dependencias entre estados no tiene ciclos y está dibujado
- cada bloque de proveedor usa identidad federada y acotada
- nadie aplica a mano con credenciales personales
- hay planificación programada por estado y alerta por antigüedad
- el plan avisa de destrucciones y recreaciones
- los entornos efímeros se crean también en el segundo proveedor

Y el cierre que enlaza con la clase siguiente: con la infraestructura declarada por proveedor, queda la capa que sí es común y que puede acabar siendo una flota difícil de gobernar: varios clústeres en varios proveedores, con las mismas políticas. Es la materia de la clase 164.

Ejemplo trabajado

CloudShop declara infraestructura en dos proveedores. Empieza con un módulo común y un estado único, y llega a nueve estados y módulos separados tras dos incidentes que explican por qué.

El módulo común, y por qué se abandonó.

```
intento: un módulo «base de datos» con una variable de proveedor

líneas del módulo                                          410
de ellas, condicionales por proveedor                      240   (59 %)
parámetros expuestos                                        14
parámetros disponibles en el proveedor A                    41
parámetros disponibles en el proveedor B                    37
```

Catorce de cuarenta y uno: el módulo exponía la intersección, y las funciones específicas de cada proveedor quedaban inaccesibles.

Y el fallo que lo cerró:

```
al añadir una opción de copia disponible solo en A,
hubo que añadir un condicional más y una variable que en B
se ignoraba en silencio
→ alguien la usó pensando que hacía algo
→ una carga de B estuvo 3 meses sin la retención que creía tener
```

líneas totales	410	180 + 165
condicionales por proveedor	240	0
parámetros expuestos	14	41 y 37
variables ignoradas en silencio	3	0
prueba de interfaz equivalente	no	sí

Y la prueba de interfaz, que se ejecuta en cada cambio:

```
comprueba que los dos módulos aceptan las mismas 9 entradas
y devuelven las mismas 5 salidas
divergencias detectadas en 8 meses                           4
  → todas, alguien añadiendo una entrada a uno solo
```

El estado único, y los dos incidentes.

```
situación inicial   un estado por entorno, abarcando los dos proveedores
recursos en el estado de producción                        1.180
```

```
tiempo de planificación                                    6 min 40
```

Incidente 1: un proveedor caído bloquea el otro.

```
10:20  incidencia en el segundo proveedor: su API no responde
10:22  hay que aplicar un cambio urgente en el PRIMERO
10:22  el plan falla: no puede leer el estado de los recursos de B
10:22  no se puede aplicar nada, en ninguno de los dos
11:40  se resuelve la incidencia de B

tiempo sin poder cambiar nada                              1 h 18
cambio urgente que esperaba                  una regla de cortafuegos
```

Incidente 2: un aplicado erróneo con radio de daño enorme.

```
un cambio en el módulo de red se aplicó sobre el estado de producción
el plan proponía destruir y recrear 3 subredes
nadie leyó el plan completo: tenía 340 líneas
recursos destruidos                                            41
tiempo de recuperación                                      2 h 10
```

El reparto resultante.

```
estados, tras el reparto                                        9
  proveedor A: cimientos, red, datos, plataforma, 3 servicios
  proveedor B: cimientos, red y las 3 cargas en un estado por carga

recursos por estado, mediana                                  110
tiempo de planificación, mediana                           41 s
radio de daño del mayor                          red de un entorno
```

estados	3	9
estados que abarcan dos proveedores	3	0
tiempo de planificación	6 min 40	41 s
un proveedor caído bloquea el otro	sí	no
líneas del plan de un cambio típico	340	28
recursos destruidos por error	41 (una vez)	0

Y la protección añadida sobre los estados críticos:

```
cimientos, red y datos
  protección contra destrucción en los recursos con datos
  aprobación de dos personas para aplicar
  y aviso destacado en el plan cuando hay destrucción y recreación

planes con destrucción detectados en 8 meses                    7
  de ellos, intencionados                                       2
  de ellos, errores detenidos por el aviso                      5
```

Cinco destrucciones evitadas por un aviso que antes se perdía entre trescientas cuarenta líneas.

Los datos entre estados.

La primera versión leía el estado ajeno directamente:

```
lecturas de estado ajeno                                       14
roturas al reorganizar un estado                                3
  → un recurso renombrado internamente rompía a dos consumidores
```

Y el cambio a salidas publicadas:

lo que expone	todo el estado	5-9 valores
roturas al reorganizar el interior	3	0
permiso necesario	lectura del fichero	lectura de las salidas
versionado	no	sí

La dependencia entre proveedores.

El caso concreto: un registro de nombres en A que apunta al balanceador de B.

```
mal   un estado que abarcara los dos, para poder referenciarlo
      → era exactamente lo que causó el incidente 1

bien  el estado de B publica la dirección del balanceador
      la canalización la toma y la pasa como variable al estado de A
      orden: B primero, A después

etapas de la canalización                                       6
```

```
orden declarado                                              sí
reanudación desde la etapa fallida                           sí
dependencias entre estados                                   11
ciclos en el grafo                                            0
  → hubo 1, resuelto fijando el rango de red de antemano en vez
    de generarlo
```

Las credenciales y la deriva.

```
identidad por proveedor                    clave estática   federada  clase 159
acotada por rama y entorno                         no             sí
aplicados a mano con credenciales personales    4 / mes            0
planificación programada por estado                no        cada 6 h, 9 estados
alerta por antigüedad de planificación             no             sí
deriva detectada en 8 meses                         —             19
  de ellas, cambios manuales                        —              4
  de ellas, campos de otros controladores           —             15 → excluidos
```

Y el estado propio, guardado donde corresponde:

```
estado de B guardado en                          A              B
qué pasaba si A caía                  no se podía arreglar B   nada
cifrado, versionado y con bloqueo               parcial          sí, los 9
acceso al fichero de estado                     4 equipos       1 equipo por estado
```

A los ocho meses.

```
módulos comunes multiproveedor                    1             0
módulos por proveedor con interfaz igual          0             2 pares
variables ignoradas en silencio                   3             0
estados                                           3             9
estados que abarcan dos proveedores               3             0
tiempo de planificación                        6 min 40         41 s
líneas del plan de un cambio típico             340             28
destrucciones erróneas detenidas por aviso        0             5
lecturas de estado ajeno                         14             0
ciclos en el grafo de dependencias                1             0
aplicados a mano                               4 / mes           0
estados con planificación programada            0 de 3         9 de 9
```

La lección que esta clase traslada a la parte 13: el módulo común exponía catorce parámetros de los cuarenta y uno disponibles y aceptaba en silencio variables que en un proveedor no hacían nada, lo que dejó una carga tres meses sin la retención que creía tener. Y el reparto del estado no se hizo por elegancia: se hizo porque una incidencia en un proveedor impidió durante setenta y ocho minutos aplicar un cambio urgente en el otro, que es exactamente el acoplamiento que un estado compartido introduce sin que nadie lo declare.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-13-multicloud-hybrid-disaster-recovery/163-terraform-multi-provider-y-separacion-de-estados/lab.py
```

El laboratorio selecciona el motor de práctica iac 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 iac-multicloud 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 plan reproducible sin secretos ni cambios inesperados. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye iac-multicloud 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 módulo multiproveedor se llena de condicionales y expone pocas opciones	Se intentó abstraer recursos con formas distintas	Un módulo por proveedor con la misma interfaz de entradas y salidas, y una prueba que compruebe que siguen siendo equivalentes.
Una variable se acepta y no hace nada en uno de los proveedores	El módulo común la ignora en silencio	Módulos separados que fallen ante una entrada desconocida, en vez de ignorarla.
Una incidencia en un proveedor impide aplicar cambios en el otro	Un estado abarca los dos	Reparte el estado por proveedor siempre, y guarda el estado de cada uno en su propio proveedor.
Un aplicado erróneo destruye decenas de recursos	El estado abarca demasiado y el plan es tan largo que nadie lo lee	Reparte por radio de daño y ciclo de vida, protege los recursos con datos y destaca las destrucciones en el plan.
Reorganizar un estado rompe a otros	Los consumidores leen el estado ajeno completo	Publica salidas con nombres estables y versionadas, y lee solo eso.
Un estado deja de planificarse y nadie se entera	Ley 13: no planificar no produce error	Planificación programada por estado con alerta por antigüedad de la última ejecución correcta.

■ Seguridad, ética y costo

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

■ Preguntas de comprobación

1. ¿Por qué falla un módulo común para varios proveedores y qué lo sustituye?
2. ¿Cuáles son los cuatro criterios para repartir el estado y cuál es innegociable?
3. ¿Por qué el estado de un proveedor debe guardarse en ese proveedor?
4. ¿Qué tres formas hay de pasar datos entre estados y cuál se recomienda entre equipos?
5. ¿Cómo se resuelve una dependencia entre proveedores sin un estado común?

Referencias

- HashiCorp (2025). Terraform: multiple provider configurations and aliases — varios proveedores en una misma configuración. <https://developer.hashicorp.com/terraform/language/providers/configuration>
- HashiCorp (2025). State: remote backends, locking and separation — almacenamiento, bloqueo y reparto del estado. <https://developer.hashicorp.com/terraform/language/state/remote>
- HashiCorp (2025). Module composition and interfaces — módulos con interfaces estables en vez de módulos condicionales. <https://developer.hashicorp.com/terraform/language/modules/develop/composition>
- Brikman, Y. (2022). Terraform: Up & Running, caps. 3 y 8 — reparto de estados y dependencias entre ellos. <https://www.terraformupandrunning.com/>
- Open Policy Agent (2025). Policy checks on infrastructure plans — análisis del plan antes de aplicar. <https://www.openpolicyagent.org/docs/latest/terraform/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 13 en PDF · Manual integral

Anterior	Índice	Siguiente
← 162 · Observabilidad y operación entre proveedores	Parte 13 · Programa	164 · Flotas Kubernetes y políticas comunes →

Evaluación — 163 Terraform multi-provider y separación de estados

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 iac-multicloud 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.

164 — Flotas Kubernetes y políticas comunes

Parte: 13 — Multi-cloud, híbrido, migración y recuperación Nivel: experto · Horas estimadas: 4 Laboratorio: kubernetes · Estado: EXECUTABLECORE

Propósito

Gobernar varios clústeres repartidos entre proveedores, regiones y entornos, cuando el coste de operarlos se multiplica por su número salvo que algo los mantenga uniformes. La clase separa los motivos legítimos para tener muchos de la proliferación accidental, fija qué debe ser idéntico en todos y qué puede variar, y desarrolla los dos problemas que definen una flota: el coste fijo de cada clúster, que hace caros los pequeños, y las actualizaciones, donde la aspiración de tener la misma versión en todas partes es imposible y lo alcanzable es una ventana acotada.

Resultados de aprendizaje

Al finalizar podrás:

1. Justificar cada clúster por un motivo, y reconocer la proliferación accidental.
2. Separar lo que debe ser idéntico en la flota de lo que puede variar.
3. Aplicar configuración y políticas a un conjunto seleccionado por etiquetas.
4. Gestionar actualizaciones con una ventana de versiones, no con una versión única.
5. Calcular el coste fijo por clúster y decidir con él.

Conceptos centrales

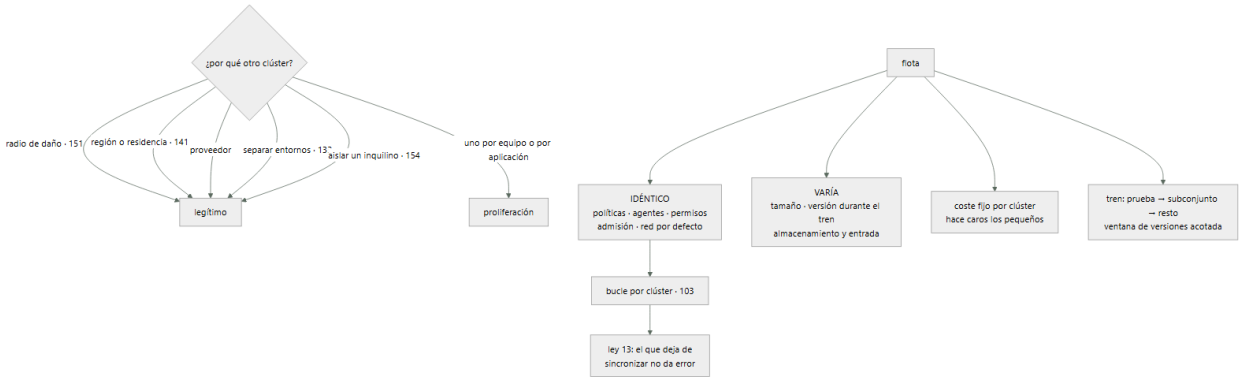
Concepto	Comprensión verificable
flota	Conjunto de clústeres gobernados como una unidad, con configuración y políticas comunes aplicadas a grupos seleccionados.
coste fijo por clúster	Lo que cuesta un clúster antes de ejecutar nada: plano de control, agentes, cargas de sistema y su parte de operación.
ventana de versiones	Diferencia máxima admitida entre la versión más antigua y la más reciente de la flota. Sustituye a la versión única, que es inalcanzable.
tren de actualización	Orden por el que una versión recorre la flota: clúster de prueba, un subconjunto y el resto.
línea base	Conjunto de recursos que existe en todos los clústeres: políticas, agentes, permisos y reglas de admisión.
colocación explícita	Decidir en qué clúster corre cada carga de forma declarada, en vez de repartirla automáticamente entre clústeres.

Modelo mental

Multi-cloud es una decisión de negocio y riesgo; duplicar cada componente entre proveedores rara vez es la forma más confiable o económica de lograrla.

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 hay muchos, y cuándo sobran

Los motivos legítimos para tener varios clústeres, cada uno con su equivalente en este programa:

```
RADIO DE DAÑO             un fallo grave afecta a una parte     clase 151
REGIÓN O RESIDENCIA       el dato no puede salir de un sitio    clase 141
PROVEEDOR                 se usa más de uno                     clase 157
SEPARACIÓN DE ENTORNOS    producción no comparte plano de control
                          con desarrollo                        clase 133
AISLAMIENTO DE INQUILINO  un cliente exige separación           clase 154
ACTUALIZACIÓN ESCALONADA  hace falta uno por delante para probar
```

Y los que no lo son:

```
uno por equipo            se resuelve con espacios de nombres y permisos
uno por aplicación        el coste fijo no lo justifica
«por si acaso»            aparece en el inventario dos años después
y el que nadie recuerda por qué existe                          ley 20
```

Y el criterio que ordena la decisión, que es el de la clase 148 aplicado aquí:

```
cada clúster tiene un coste fijo
y cada uno necesita un motivo escrito
→ si no hay motivo, se consolida
```

El coste fijo, que es lo que hace caros los clústeres pequeños:

```
plano de control gestionado                    coste mensual fijo
cargas de sistema: red, resolución de nombres,
  métricas, registro, agente de políticas      2-4 nodos equivalentes
agentes de seguridad y de observabilidad       por nodo
margen para tolerar la caída de un nodo        1 nodo como mínimo
la parte de operación: actualizaciones, ensayos,
  procedimientos y guardia                     el mayor de todos
```

Y la aritmética que decide:

```
coste fijo por clúster, orden de magnitud    equivalente a 3-5 nodos
un clúster con 4 nodos de carga útil          más del 50 % es sistema
un clúster con 40 nodos                        ~10 %
```

Y de ahí la regla práctica:

```
pocos clústeres grandes salen mucho más baratos que muchos pequeños
→ y el motivo para tener uno pequeño tiene que valer más que su
  coste fijo
→ un clúster por cliente solo se sostiene si el cliente lo paga
                                                      clase 154
```

Y una consecuencia que conviene anticipar: el número de clústeres tiende a crecer y nunca a bajar, porque nadie propone consolidar. La revisión periódica del inventario, con el motivo de cada uno, es lo único que lo contiene.

2. Qué es idéntico y qué varía

Una flota se gobierna decidiendo esto y no volviendo a discutirlo:

```
IDÉNTICO EN TODOS
  reglas de admisión: firma verificada, campos obligatorios  clases 067, 101
```

```
  políticas de red por defecto: denegar entre espacios      clase 135
  modelo de permisos y grupos                                clase 159
  agentes de observabilidad y su configuración               clase 162
  etiquetas obligatorias de dueño y entorno                  clase 142
  límites y cuotas por espacio de nombres                    clase 078
  y las cinco fronteras absolutas                            clase 144

PUEDE VARIAR
  número y tamaño de nodos
  versión, durante el paso del tren de actualización
  clase de almacenamiento y controlador de entrada
  configuración específica del proveedor
  y las cargas que corren en cada uno
```

Y lo que hace que lo idéntico siga siéndolo:

```
se declara una vez y se aplica a un CONJUNTO seleccionado por etiquetas
  «todos los de producción»
  «todos los del proveedor B»
  «todos los que sirven a clientes europeos»
→ y añadir un clúster con esas etiquetas lo incorpora solo
```

Y la advertencia que arrastra, porque esto es la clase 103 multiplicada:

```
cada clúster tiene su bucle de reconciliación
→ un clúster que deja de sincronizar NO da ningún error   ley 13
→ y se queda con la configuración de hace semanas, incluidas
  las políticas de seguridad
```

Y por eso la señal obligatoria es la misma de la clase 103, ahora por clúster:

```
antigüedad de la última reconciliación correcta, POR CLÚSTER
y un panel que muestre, para cada uno, si está al día
```

Y una comprobación que va más allá y que casi nadie hace:

```
no basta con que el bucle diga que aplicó
hay que COMPROBAR EL RESULTADO en cada clúster
  ¿existe la regla de admisión?
  ¿está la política de red por defecto?
  ¿se rechaza una imagen sin firmar?
→ son las pruebas negativas de la clase 144, ejecutadas en TODOS
```

Y el motivo: un bucle puede estar sincronizado y la política no estar activa, porque un componente falló al arrancar o porque alguien creó una excepción local.

3. Actualizar una flota

Aquí está la carga operativa real, y empieza por aceptar algo:

```
LA MISMA VERSIÓN EN TODAS PARTES ES INALCANZABLE
  los proveedores publican y retiran versiones en calendarios distintos
  las actualizaciones del plano de control gestionado se imponen
  y una flota de nueve clústeres nunca está toda igual
```

Lo alcanzable es acotar la diferencia:

```
VENTANA DE VERSIONES
  «entre la más antigua y la más reciente, como máximo una versión menor»
→ y eso sí se puede vigilar y hacer cumplir
```

Y el orden en que una versión recorre la flota:

```
TREN DE ACTUALIZACIÓN
  1. un clúster de prueba, sin tráfico real            días
  2. un clúster de producción de bajo riesgo           1 semana
  3. el resto, por grupos                              2-3 semanas
→ y entre etapas, tiempo suficiente para que aparezca lo que falla
```

Y lo que hay que comprobar en cada etapa, más allá de que arranque:

```
las cargas siguen funcionando: canario y objetivos     clases 102, 126
las API que se usan siguen existiendo
  → las versiones retiran API, y una carga que las use dejará de aplicar
los complementos y controladores son compatibles
y los agentes de observabilidad y seguridad siguen reportando
```

Y la comprobación que ahorra la mayoría de los disgustos:

```
antes de actualizar, buscar en TODOS los clústeres el uso de API
que la versión nueva retira
→ se puede hacer con las herramientas del propio orquestador
→ y lo que aparece siempre son manifiestos antiguos y complementos
  de terceros
```

Y la parte que se olvida:

```
los nodos se actualizan aparte del plano de control
→ y la diferencia entre ambos tiene un límite admitido
→ superarlo produce fallos difíciles de diagnosticar
```

Y el ensayo correspondiente, del catálogo de la clase 131:

```
actualizar un clúster de prueba y comprobar que las cargas
sobreviven al drenaje de nodos
→ y ahí se descubre quién no tolera que le muevan las instancias:
  ausencia de presupuesto de interrupción, terminación sucia
  o estado local escondido                          clases 079, 146
```

Y una cifra que conviene tener para dimensionar el esfuerzo:

```
actualizaciones al año por clúster                   3-4
clústeres                                            9
actualizaciones al año en total                      27-36
→ es la carga que justifica automatizar el tren entero
```

4. Colocar cargas y medir la flota

Dónde corre cada carga conviene decidirlo de forma explícita:

```
COLOCACIÓN EXPLÍCITA
  la declaración dice a qué clúster o grupo va              clase 103
  + previsible, auditable y fácil de razonar
  – hay que decidirlo

REPARTO AUTOMÁTICO ENTRE CLÚSTERES
  un componente decide dónde colocar según capacidad
  + aprovecha mejor
  – la carga puede moverse sin que nadie lo espere
  – los datos NO se mueven con ella                         ley 21
  – y el diagnóstico se complica: «¿dónde está corriendo esto?»
```

Y la posición honesta: el reparto automático entre clústeres rara vez compensa, porque el estado y la latencia atan la carga a un sitio, y porque el aprovechamiento se consigue mejor con menos clústeres y más grandes.

Donde sí encaja algo parecido:

```
la misma carga desplegada en VARIOS clústeres a la vez, con reparto
de tráfico por delante
→ eso no es colocación automática: es replicación declarada
→ y es la base del activo-activo de la clase 168
```

Lo que hay que medir de una flota, además de lo de cada clúster:

```
clústeres, y el motivo escrito de cada uno
antigüedad de la última reconciliación, por clúster
resultado de las pruebas negativas, por clúster
versión del plano de control y de los nodos, y la ventana
clústeres fuera de la ventana de versiones
coste fijo por clúster y proporción sobre su coste total
cargas por clúster y aprovechamiento
y clústeres sin ninguna carga                             ley 20
```

La última suele dar sorpresas: un clúster vacío sigue costando su parte fija.

Y el gobierno que mantiene la flota acotada:

```
crear un clúster exige un motivo del catálogo y un dueño
cada clúster se revisa al menos una vez al año
y el que no tiene motivo vigente se consolida o se apaga
```

Y la lista de comprobación de la clase:

- ■ cada clúster tiene un motivo escrito y un dueño
- ■ no hay clústeres por equipo ni por aplicación sin más motivo
- ■ está calculado el coste fijo por clúster
- ■ está escrito qué es idéntico en toda la flota y qué puede variar

- lo idéntico se aplica por selección de etiquetas, no clúster a clúster
- hay alerta por antigüedad de reconciliación, por clúster
- las pruebas negativas se ejecutan en TODOS, no solo en uno
- existe ventana de versiones declarada y se vigila
- hay tren de actualización con etapas y tiempo entre ellas
- se busca uso de API retiradas antes de cada actualización
- se ensaya el drenaje de nodos y sus efectos en las cargas
- la colocación de cargas es explícita
- se revisa el inventario y se consolida lo que sobra

Y el cierre que enlaza con la clase siguiente: hasta aquí, todo ocurre en nubes públicas. Cuando parte del sistema vive en instalaciones propias o en el borde —con conectividad peor, hardware que hay que mantener y sitios sin nadie que los atienda— aparecen problemas distintos, y son la materia de la clase 165.

Ejemplo trabajado

CloudShop tiene nueve clústeres repartidos entre dos proveedores. El ejercicio empieza inventariando por qué existe cada uno y termina con seis, tras descubrir que uno llevaba seis semanas sin recibir configuración.

El inventario, con el motivo de cada uno.

```
clúster                  motivo declarado            ¿legítimo?
A-produccion-eu          producción principal            sí
A-produccion-eu-2        radio de daño                   sí
A-preproduccion          separación de entornos          sí
A-desarrollo             separación de entornos          sí
A-datos                  «el equipo de datos quería
                          el suyo»                       no
A-antiguo                nadie lo recordaba              no
B-produccion-eu          proveedor y residencia          sí
B-cliente-x              aislamiento de inquilino        sí
B-pruebas                pruebas de portabilidad · 158   sí
```

Y el coste fijo, calculado:

```
coste fijo por clúster, medido
  plano de control gestionado                    ~90 €/mes
  cargas de sistema (3 nodos equivalentes)      ~310 €/mes
  agentes por nodo                               ~40 €/mes
  margen de un nodo                             ~100 €/mes
                                                ■■■■■■■■■■
                                                ~540 €/mes

coste fijo de los 9                            ~4.860 €/mes
```

Y los dos sin motivo:

```
A-datos      4 nodos, 2 cargas que caben en el principal
             coste total 810 €/mes, de los que 540 son fijos   → 67 %
A-antiguo    2 nodos, 0 cargas desde hacía 8 meses             ley 20
             coste 620 €/mes

consolidados y apagado
ahorro                                              1.430 €/mes
clústeres                                                 9 → 7
```

Y después, dos más:

```
B-pruebas    se sustituyó por un clúster efímero creado por la
             prueba de portabilidad y destruido al terminar  clase 104
             coste                          540 €/mes → 12 €/mes
A-preproduccion y A-desarrollo   se unieron con espacios de nombres
             y cuotas separadas; el motivo era separación de entornos,
             y desarrollo no necesita plano de control propio
             coste                          1.080 €/mes → 540 €/mes

clústeres finales                                          6
coste fijo total                       4.860 €/mes → 3.010 €/mes
```

El clúster que llevaba seis semanas sin sincronizar.

Al montar el panel de flota:

```
clúster                  última reconciliación correcta
A-produccion-eu          hace 4 min
```

```
A-produccion-eu-2          hace 6 min
A-preproduccion            hace 3 min
B-produccion-eu            hace 5 min
B-cliente-x                HACE 42 DÍAS
A-antiguo                  hace 187 días
```

Y qué significaba en el clúster del cliente:

```
cambios de política no aplicados                          14
  de ellos, de seguridad                                   6
  incluida la regla de admisión de firma verificada  clases 067, 101
versión de los agentes de observabilidad             3 versiones atrás
causa    un permiso caducado del agente de reconciliación
tiempo sin que nadie lo supiera                          42 días
```

Seis cambios de política de seguridad sin aplicar en el clúster de un cliente que exigía aislamiento, y ningún error en ninguna parte. Es la ley 13, en su versión de flota.

```
alerta por antigüedad de reconciliación         no       sí, por clúster
umbral                                           —          30 min
pruebas negativas ejecutadas en                1 clúster   6 clústeres
fallos encontrados al ejecutarlas en todos       —             4
  → 3 en B-cliente-x, por lo anterior
  → 1 en A-produccion-eu-2: una excepción local que nadie recordaba
```

Y el cuarto es el que justifica comprobar el resultado y no solo el estado del bucle: el bucle decía que estaba sincronizado y la política no estaba activa, por una excepción creada a mano meses antes.

Las versiones.

```
situación inicial
  versiones distintas del plano de control                  4
  diferencia entre la más antigua y la más nueva         3 menores
  clústeres con nodos fuera del margen admitido             2
```

```
ventana declarada                          no había     1 versión menor
clústeres fuera de la ventana                3             0
tren de actualización                      no había      3 etapas
tiempo entre etapas                          —          7 días
actualizaciones al año                      ~24           ~18
automatizadas                                0 %          90 %
```

Y la comprobación previa, que se añadió al tren:

```
búsqueda de uso de API retiradas, en los 6 clústeres
  primera ejecución                                hallazgos: 31
    manifiestos antiguos de 2 cargas                          18
    dos complementos de terceros                              11
    un trabajo programado                                      2
  actualizaciones que habrían fallado sin corregirlo           2
```

Y el ensayo de drenaje, que reveló lo previsible:

```
drenar un nodo a propósito, en preproducción
  cargas sin presupuesto de interrupción                    4
  cargas con terminación sucia                              2   clase 146
  cargas con estado local                                   1
tras corregirlas, drenaje sin efecto observable
```

La colocación.

```
propuesta inicial   un componente que repartiera cargas entre clústeres
                    según capacidad
análisis            los datos de cada carga están en un proveedor
                    y una región concretos                  ley 21
                    mover la carga sin los datos multiplica la latencia
                    y el coste de salida                   clase 161
decisión            colocación explícita, declarada         clase 103

excepción           la carga del flujo de compra se despliega en los
                    dos clústeres de producción de A, con reparto
                    de tráfico por delante
                    → replicación declarada, no colocación automática
```

Lo que se mide de la flota.

```
clústeres                                  9             6
```

```
con motivo escrito y dueño                3 de 9        6 de 6
coste fijo total                      4.860 €       3.010 €
proporción del coste que es fijo        34 %          19 %
clústeres sin ninguna carga               1             0
reconciliación vigilada                  no            sí
el más desactualizado                 42 días       18 min
pruebas negativas en todos               no            sí
ventana de versiones                  3 menores     1 menor
clústeres fuera de la ventana             3             0
revisión anual del inventario            no            sí
```

La lección que esta clase traslada a la parte 13: dos de los nueve clústeres no tenían ningún motivo y costaban mil cuatrocientos treinta euros al mes, de los que la mayor parte era coste fijo: un clúster pequeño gasta más en existir que en trabajar. Y el hallazgo grave no fue de coste: el clúster de un cliente que exigía aislamiento llevaba cuarenta y dos días sin recibir configuración, incluidas seis políticas de seguridad, sin que nada diera error; y una de las cuatro comprobaciones que se ejecutaron después reveló que un bucle sincronizado no garantiza que la política esté activa.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-13-multicloud-hybrid-disaster-recovery/164-flotas-kubernetes-y-politicas-comunes/lab.py
```

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

Evidencia esperada

El artefacto principal es manifiestos declarativos con estado observado. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye fleet-kubernetes 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 coste de la plataforma crece más deprisa que la carga	Muchos clústeres pequeños, cada uno con su coste fijo	Calcula el coste fijo por clúster, exige un motivo escrito para cada uno y consolida lo que no lo tenga.
Un clúster tiene políticas antiguas y nadie lo sabe	Ley 13: su bucle de reconciliación dejó de sincronizar y eso no produce error	Alerta por antigüedad de reconciliación por clúster, y panel de flota con el estado de todos.
El bucle dice que está al día y una política no está activa	Un componente falló al arrancar o hay una excepción local	Ejecuta las pruebas negativas en todos los clústeres, no solo en uno; comprueba el resultado, no el estado del bucle.
Una actualización rompe cargas que llevaban años funcionando	La versión nueva retira API que esos manifiestos usan	Busca uso de API retiradas en toda la flota antes de actualizar, y avanza por un tren con etapas y tiempo entre ellas.
Se persigue tener la misma versión en todos y nunca se consigue	Los calendarios de los proveedores no coinciden y las actualizaciones gestionadas se imponen	Declara una ventana de versiones y vigila quién se sale de ella.
Nadie sabe en qué clúster corre una carga	Se reparte automáticamente entre clústeres	Colocación explícita y declarada; si hace falta redundancia, despliega en varios a la vez con reparto de tráfico por delante.

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 motivos justifican un clúster más y cuáles no?
2. ¿Por qué un clúster pequeño es caro, y qué proporción de su coste es fija?
3. ¿Qué debe ser idéntico en toda la flota y cómo se aplica?
4. ¿Por qué no basta con que el bucle diga que está sincronizado?
5. ¿Por qué la misma versión en todas partes es inalcanzable y qué se declara en su lugar?

Referencias

- Kubernetes (2025). Version skew policy — diferencias admitidas entre plano de control y nodos. <https://kubernetes.io/releases/version-skew-policy/>
- Kubernetes (2025). Deprecated API migration guide — detectar uso de API retiradas antes de actualizar. <https://kubernetes.io/docs/reference/using-api/deprecation-guide/>
- Argo CD (2025). ApplicationSet: cluster generators — aplicar configuración a conjuntos seleccionados por etiquetas. <https://argo-cd.readthedocs.io/en/stable/operator-manual/applicationset/Generators-Cluster/>
- Google Cloud (2025). Fleet management and policy consistency — flota, línea base y políticas comunes. <https://cloud.google.com/kubernetes-engine/fleet-management/docs>
- CNCF (2025). Multicluster management patterns — colocación explícita frente a reparto automático. <https://www.cncf.io/reports/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 13 en PDF · Manual integral

Evaluación — 164 Flotas Kubernetes y políticas comunes

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 fleet-kubernetes 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.

165 — Nube híbrida, edge y conectividad privada

Parte: 13 — Multi-cloud, híbrido, migración y recuperación Nivel: experto · Horas estimadas: 4 Laboratorio: hybrid · Estado: EXECUTABLECORE

Propósito

Diseñar para la parte del sistema que no vive en una nube pública: instalaciones propias y sitios remotos donde el hardware es tuyo, la capacidad no crece sola, la conexión es peor o intermitente y no hay nadie a quien pedirle que toque un cable. La clase interroga los motivos con el método de la clase 157, desarrolla el problema central —funcionar sin conexión y reconciliar al volver, que es la ley 21 en su forma más dura— y trata los dos riesgos que no existen en la nube: una actualización que deja un sitio inaccesible y un disco que alguien se puede llevar.

Resultados de aprendizaje

Al finalizar podrás:

1. Interrogar los motivos para tener algo fuera de la nube pública.
2. Situar cada emplazamiento por quién puede tocarlo y en cuánto tiempo.
3. Diseñar para operar sin conexión y reconciliar al recuperarla.
4. Actualizar sin poder acceder al sitio si la actualización falla.
5. Proteger un equipo al que alguien tiene acceso físico.

Conceptos centrales

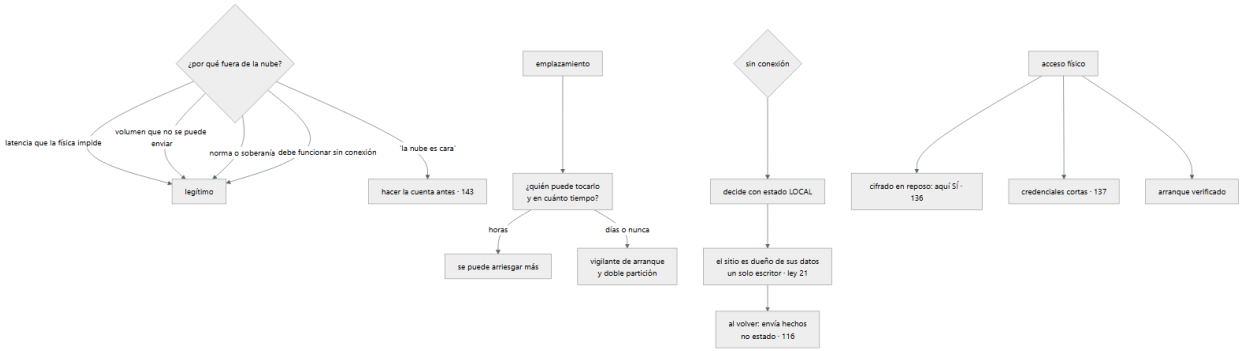
Concepto	Comprensión verificable
operación sin conexión	Capacidad de seguir prestando servicio cuando el enlace con el centro se pierde, decidiendo con estado local.
propiedad del dato por emplazamiento	Cada sitio es el escritor de sus propios datos. Es lo que evita conflictos al reconciliar.
tiempo hasta que alguien lo toca	Cuánto se tarda en tener a una persona delante del equipo. Decide qué se puede arriesgar en una actualización.
vigilante de arranque	Mecanismo que revierte a la versión anterior si la nueva no logra funcionar. Es lo que impide dejar un sitio inaccesible.
guardar y reenviar	Acumular telemetría y hechos localmente y enviarlos cuando haya enlace, en vez de perderlos.
acceso físico	Amenaza que no existe en una nube pública: alguien puede llevarse el equipo o su disco.

Modelo mental

Multi-cloud es una decisión de negocio y riesgo; duplicar cada componente entre proveedores rara vez es la forma más confiable o económica de lograrla.

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 algo vive fuera

Los motivos, con la pregunta de siempre:

```
LATENCIA QUE LA FÍSICA IMPIDE
  un control industrial que necesita responder en milisegundos
  un terminal de venta que no puede esperar 80 ms por cada lectura
  ¿qué pasa si no?   el proceso no funciona
  → sólido, y se comprueba con números, no con impresiones

VOLUMEN QUE NO SE PUEDE ENVIAR
  vídeo, sensores a alta frecuencia, imágenes médicas
  ¿qué pasa si no?   el enlace no da, o el coste de salida es
                     prohibitivo                          clase 161
  → sólido, y la respuesta suele ser procesar en el sitio y enviar
    el resultado, que es la regla número uno de la clase 161

NORMA O SOBERANÍA
  datos que no pueden salir de un edificio o de un país  clase 141
  → sólido

DEBE FUNCIONAR SIN CONEXIÓN
  una tienda no puede dejar de cobrar porque se caiga la línea
  → sólido, y es el que define el diseño: apartado tercero

INVERSIÓN EXISTENTE CON VIDA ÚTIL
  hay hardware comprado que aún sirve
  → legítimo, y es temporal: tiene fecha

«LA NUBE ES CARA»
  ¿comparado con qué?   con el coste total: hardware, espacio,
                        energía, repuestos, personal, renovación
  → hay casos donde es cierto, y la cuenta hay que hacerla entera
                                                       clase 143
```

Y el criterio que ordena la decisión, igual que en la clase 157:

```
cada emplazamiento fuera de la nube tiene un motivo escrito
y lo que no lo tenga, se consolida
```

El espectro, que decide casi todo lo demás:

```
UBICACIONES DE BORDE DEL PROVEEDOR
  el proveedor pone y mantiene el hardware; tú despliegas
  → operable casi como la nube

EQUIPO GESTIONADO EN TUS INSTALACIONES
  el proveedor lo mantiene y actualiza; está en tu edificio
  → operación conocida, y depende de un contrato

HARDWARE PROPIO EN UN CENTRO DE DATOS
  → capacidad fija, repuestos, y personal

HARDWARE PROPIO EN UN SITIO SIN NADIE
  un armario en una tienda, un cuadro en una fábrica
  → el caso duro, y el que define esta clase
```

Y la variable que hay que anotar para cada emplazamiento:

```
¿QUIÉN PUEDE TOCARLO Y EN CUÁNTO TIEMPO?
  minutos       alguien en el edificio, formado
  horas         un técnico que se desplaza
  días          hay que contratar a alguien
  nunca         no hay acceso posible
→ y de esa respuesta depende cuánto se puede arriesgar en cada cambio
```

2. Funcionar sin conexión

Es el requisito que define el diseño, y su consecuencia es directa:

```
si el sitio debe seguir funcionando sin enlace,
el sitio decide con estado LOCAL
→ y por tanto ESCRIBE localmente
→ y entonces, ¿quién es el dueño de ese dato?          ley 21
```

Y la respuesta que evita el problema de la clase 149:

```
CADA EMPLAZAMIENTO ES DUEÑO DE SUS PROPIOS DATOS
  las ventas de esta tienda las escribe esta tienda
  el inventario de este almacén lo escribe este almacén
→ un solo escritor por dato, aunque haya 400 sitios
→ y el centro es un CONSUMIDOR que agrega, no una autoridad
```

Y lo que se replica en cada sentido:

```
del centro al sitio    catálogo, precios, reglas, configuración
                       → lectura; el sitio no los modifica
del sitio al centro    hechos: ventas, movimientos, mediciones
                       → un sentido, con hechos y no con estado
                                                  clases 115, 161
```

Y lo que hay que resolver para lo que no se puede partir:

```
el inventario de un producto que se vende en la tienda y en la web
  → dos escritores: el problema real

salidas honestas
  reservar un cupo por tienda: cada una escribe su cupo   → partir
  aceptar sobreventa y compensar                           clase 149
  o exigir conexión para esa operación concreta
→ y hay que decidirlo por operación, no en general
```

Al recuperar la conexión, lo que ocurre y cómo se gestiona:

```
el sitio tiene una cola de hechos acumulados             clase 116
  con identificadores propios y orden local
se envían, y el centro los aplica de forma idempotente
y el centro envía lo que cambió mientras tanto
```

Y los tres problemas que aparecen siempre:

```
AVALANCHA AL RECONECTAR
  400 sitios recuperan a la vez tras un corte regional
  → y todos envían su cola
  → hace falta variación aleatoria y limitación de caudal  clases 111, 130

COLA QUE CRECE SIN LÍMITE
  un sitio que lleva dos semanas sin enlace
  → la cola debe estar acotada y decidir qué se descarta
  → normalmente: nada de lo que sea dinero, y sí la telemetría

RELOJES
  un sitio sin conexión pierde sincronía                   clase 149
  → el orden se decide con contadores locales, no con marcas de tiempo
```

Y una decisión que conviene tomar de antemano:

```
¿qué NO se puede hacer sin conexión?
  y qué se le dice a la persona que está delante del terminal
→ un mensaje claro vale más que un fallo silencioso
```

3. Actualizar sin poder ir

Aquí está el riesgo que no existe en la nube:

```
una actualización que falla al arrancar deja el sitio inaccesible
y no hay consola remota, ni se puede recrear la instancia
```

Y el modelo que funciona es el de la clase 103, con dos añadidos:

```
EL SITIO TIRA, no se le empuja
  un agente local consulta qué versión debe tener y se actualiza
  → funciona con conexiones intermitentes y detrás de cortafuegos
  → y no hace falta alcanzar 400 sitios desde fuera

VIGILANTE DE ARRANQUE
  si la versión nueva no consigue funcionar en N minutos,
  se vuelve a la anterior automáticamente
  → doble partición o imagen anterior conservada
  → es lo único que impide dejar un sitio muerto

DESPLIEGUE POR GRUPOS
  1 sitio → 5 sitios → 10 % → resto                      clase 102
  con tiempo entre etapas suficiente para que aparezcan los fallos
  → y con criterio de parada automático
```

Y las comprobaciones que el vigilante debe hacer antes de dar por buena una versión:

```
¿arranca el sistema?
¿arranca la aplicación?
¿responde su comprobación de salud profunda?          clase 151
¿puede hablar con el centro, o al menos funcionar sin él?
→ y solo entonces se marca la versión como buena
```

Y dos precauciones más:

```
NUNCA ACTUALIZAR EL CONJUNTO ENTERO A LA VEZ, aunque parezca trivial
  incluida la configuración: un valor mal puesto tiene el mismo
  efecto que una imagen rota

Y UN CAMINO MANUAL QUE FUNCIONE SIN RED
  una forma de que alguien en el sitio restaure una versión conocida
  → un medio de arranque, un botón, un procedimiento de una página
  → y probado con alguien que no sea quien lo escribió    clase 128
```

Y el ensayo correspondiente, del catálogo de la clase 131:

```
desplegar a propósito una versión que no arranca en un sitio de prueba
  ¿revierte el vigilante?
  ¿en cuánto tiempo?
  ¿queda registro y avisa?
```

La capacidad, que tampoco se comporta como en la nube:

```
no hay autoescalado: lo que hay es lo que hay
→ se dimensiona para el pico conocido, con margen        clase 129
→ y la degradación bajo sobrecarga hay que diseñarla:
  descartar, encolar o reducir funciones                clase 130

y el fallo de hardware no se recrea: hay que tener repuestos
→ o aceptar que ese sitio queda fuera hasta que llegue uno
→ y eso es una cifra que hay que escribir, no una sorpresa
```

4. Acceso físico y visibilidad

El acceso físico cambia el modelo de amenazas, y es donde el cifrado en reposo deja de ser una casilla:

```
en una nube pública   el disco no se lo puede llevar nadie   clase 136
en un armario de una tienda   sí
```

Y las defensas, con la honestidad de la clase 136:

```
CIFRADO EN REPOSO      aquí sí protege del escenario real
  y la clave no puede estar en el mismo disco
  → módulo de plataforma en el equipo, o clave que se obtiene
    del centro al arrancar y no se persiste
  → y entonces, sin conexión no arranca: hay que decidirlo

ARRANQUE VERIFICADO    que no se pueda sustituir el sistema

CREDENCIALES CORTAS    lo que haya en el equipo debe caducar   clase 137
  → una credencial robada de un terminal debe servir horas, no años
```

```
  → y la identidad del sitio debe poder revocarse desde el centro

MÍNIMO PRIVILEGIO POR SITIO                                  clase 134
  el terminal de una tienda solo puede escribir lo de esa tienda
  → y eso lo comprueba el CENTRO, no el terminal

SIN PUERTOS DE ADMINISTRACIÓN EXPUESTOS
  y acceso remoto solo iniciado desde el sitio hacia fuera
```

Y la regla que resume el modelo:

```
el equipo del emplazamiento NO ES DE CONFIANZA
→ todo lo que envía se valida en el centro
→ y comprometer un sitio debe alcanzar solo a ese sitio    clase 133
```

La visibilidad, con el ancho de banda como restricción:

```
no se puede enviar telemetría como desde la nube             clase 162
→ agregar mucho más agresivamente en el sitio
→ guardar y reenviar: acumular y enviar cuando haya enlace
→ y priorizar: los errores y los hechos de negocio antes que las métricas
```

Y la alerta imprescindible, que es la ley 13 en su forma más literal:

```
un sitio que deja de reportar no genera ningún error
→ alerta por ANTIGÜEDAD del último informe, por emplazamiento
→ y distinguir «no hay enlace» de «el equipo está apagado o roto»
```

Y las cifras que se vigilan en una red de emplazamientos:

```
sitios que no reportan, y desde cuándo
sitios fuera de la ventana de versiones                      clase 164
tamaño de la cola pendiente de enviar, por sitio
tiempo medio sin conexión, por sitio
sitios con hardware en fallo y sin repuesto
y sitios sin motivo vigente para existir                     ley 20
```

Y la lista de comprobación de la clase:

- ■ cada emplazamiento tiene un motivo escrito y con fecha si es temporal
- ■ está anotado quién puede tocarlo y en cuánto tiempo
- ■ está decidido qué funciona sin conexión y qué no
- ■ cada sitio es dueño de sus datos: un solo escritor
- ■ lo que no se puede partir tiene una decisión explícita
- ■ los hechos se envían con cola acotada e idempotencia
- ■ la reconexión tiene variación aleatoria y limitación de caudal
- ■ el orden local no depende de marcas de tiempo
- ■ la actualización la tira el sitio, por grupos y con vigilante de arranque
- ■ existe un camino manual de recuperación, probado
- ■ está dimensionada la capacidad y escrita la degradación
- ■ hay política de repuestos y un plazo escrito
- ■ el disco está cifrado y la clave no vive en él
- ■ las credenciales del sitio caducan y se pueden revocar
- ■ el centro valida lo que llega y acota lo que un sitio puede tocar
- ■ hay alerta por antigüedad del último informe, por emplazamiento

Y el cierre que enlaza con la clase siguiente: con cargas en varias nubes y en emplazamientos propios, queda la pregunta que la parte lleva aplazando desde la clase 157: qué se hace cuando se pierde uno de esos sitios entero, con qué plazo y con qué pérdida. Es la materia de la clase 166.

Ejemplo trabajado

CloudShop opera 340 tiendas con terminales de venta propios. El ejercicio parte de un requisito claro —cobrar sin conexión— y termina con dos hallazgos: una actualización que dejó doce tiendas inoperativas y un disco que desapareció.

Los motivos, interrogados.

1. ```
   «los terminales deben cobrar sin conexión»
   ¿qué pasa si no?   la tienda cierra cuando falla la línea
   cortes medidos en 12 meses          41 tiendas, 3,2 h de media
   ventas perdidas si no funcionara     ~180.000 €/año
   → SÓLIDO, y define el diseño
   ```

2. ```
   «el vídeo de las cámaras no se puede subir»
   ```

```
   volumen                          2,1 TB/día por tienda
   coste de subirlo                 inviable              clase 161
   → SÓLIDO: se procesa en el sitio y se envía el resultado

3. «el hardware ya está comprado»
   vida útil restante               2 años
   → LEGÍTIMO Y TEMPORAL: con fecha

4. «la nube es cara para esto»
   cuenta completa   hardware, energía, repuestos, técnico
                     y renovación cada 4 años
   → no era cierto para el procesamiento de vídeo;
     sí para el terminal, por el motivo 1
```

El quién puede tocarlo.

```
tiendas con personal formado en el sitio                    0
tiempo hasta tener a un técnico delante, mediana       26 horas
tiempo en tiendas rurales                              3 días
→ una actualización que deje un equipo muerto cuesta,
  de media, un día de cierre de esa tienda
```

El diseño sin conexión.

```
dueño de cada dato
  ventas de la tienda                  la tienda        ley 21
  inventario de la tienda              la tienda
  catálogo y precios                   el centro (solo lectura)
  clientes y fidelización              el centro (solo lectura,
                                       con caché local)
```

Y el caso que no se podía partir:

```
inventario de un producto vendido en tienda y en la web
  dos escritores                                       conflicto

decisión   cupo por tienda
  el centro asigna un cupo a cada tienda cada noche
  la tienda escribe solo sobre su cupo
  si se agota, el terminal exige conexión para vender más

ventas bloqueadas por cupo agotado sin conexión, en 12 meses   19
→ frente a la alternativa: sobreventa y compensación
```

La reconexión, y la avalancha.

```
primer corte regional tras el despliegue
  tiendas afectadas                                       112
  duración del corte                                   41 min
  al restablecerse, todas enviaron su cola a la vez
  peticiones por segundo en el centro                   3.400
  capacidad                                             1.200
  resultado    el centro se saturó 9 minutos
```

```
variación aleatoria al reconectar              no      0-120 s
limitación de caudal por sitio                 no      50 hechos/s
cola acotada                                   no      10.000 hechos
qué se descarta al llenarse                    —       telemetría, nunca
                                                       ventas
pico al reconectar 112 tiendas              3.400/s      940/s
saturación del centro                        9 min        0
```

La actualización que dejó doce tiendas fuera.

```
06:00  se despliega una versión nueva del terminal a las 340 tiendas
06:04  12 tiendas no arrancan: un controlador de la impresora fiscal
       incompatible con el núcleo nuevo
06:04  esas tiendas no pueden cobrar
08:30  abre la primera; llamada a soporte
       (las tiendas abren a las 9:00)
09:10  se decide enviar técnicos
primera tienda recuperada                              11:20
última tienda recuperada                               siguiente día
ventas perdidas                                     ~14.000 €
```

Y los tres fallos que lo permitieron:

```
se desplegó a las 340 a la vez
no había vigilante de arranque: la versión rota se quedó
y no había camino manual: el técnico tuvo que llevar un equipo
```

```
despliegue                              todas a la vez   1 → 5 → 10 % → resto
tiempo entre etapas                          —            24 h
vigilante de arranque                        no        sí, revierte en 6 min
doble partición                              no             sí
camino manual sin red                        no        medio de arranque
                                                        y guía de 1 página
criterio de parada automático                no        > 2 % de fallos

ensayo: desplegar una versión que no arranca
  tiendas afectadas                     12 (real)        1 (ensayo)
  tiempo hasta revertir                 26 h             6 min
  intervención humana                   sí               no
```

Y el camino manual se probó con alguien que no lo escribió, según la clase 128:

```
preguntas durante la prueba                                    9
  → 4 de permisos, 3 de dónde está el medio, 2 de qué hacer si falla
tras corregir la guía                                          0
```

El disco que desapareció.

```
robo en una tienda; se llevaron el equipo del armario
contenido del disco
  ventas locales de los últimos 90 días                    sí
  datos de clientes de esas ventas                         sí
  credenciales del terminal hacia el centro                sí
cifrado en reposo                                          no
```

Y la respuesta, que fue la de la clase 137:

```
rotar         la credencial del terminal, revocada en 40 min
corregir      el mecanismo
purgar        no aplicable
revisar uso   ningún acceso con esa credencial tras el robo
```

Y las correcciones, que son la excepción honesta de la clase 136:

```
cifrado en reposo del disco                    no             sí
dónde vive la clave                             —        módulo del equipo,
                                                         no en el disco
arranque verificado                            no             sí
credencial del terminal                   permanente     caduca en 24 h,
                                                         renovada al conectar
qué puede escribir esa credencial          casi todo    solo datos de su tienda
revocación desde el centro                  manual, 40 min   inmediata
qué alcanzaría un equipo robado           datos de 90 días  datos cifrados
                                           + credencial      y credencial
                                                             caducada
```

Y la decisión que hubo que tomar y escribir:

```
si la clave la diera el centro al arrancar, un equipo sin conexión
no podría arrancar tras un corte de luz
→ y eso incumple el requisito 1
→ decisión: clave en el módulo del propio equipo, ligada al arranque
  verificado
→ compromiso aceptado: quien robe el equipo ENTERO y consiga
  arrancarlo podría leer; quien solo se lleve el disco, no
```

La visibilidad.

```
telemetría enviada por tienda            1,4 GB/día      12 MB/día
cómo                                     en bruto      agregada en el sitio,
                                                        guardar y reenviar
alerta por antigüedad del último informe    no       sí, > 30 min
distinguir sin enlace de equipo apagado     no             sí
tiendas sin reportar, detectadas en 12 meses  —            31
  de ellas, cortes de línea                   —            26
  de ellas, equipo averiado                   —             5
    → detectadas de media 4 h antes de que la tienda llamara
```

A los doce meses.

```
tiendas con operación sin conexión          340 de 340    340 de 340
```

```
escritores por dato                          2 (inventario)     1
saturación del centro al reconectar           9 min              0
despliegue                                todas a la vez     por grupos
vigilante de arranque                          no                sí
tiendas inoperativas por una actualización     12                0
camino manual probado                          no                sí
cifrado en reposo                              no           340 de 340
credenciales permanentes en tiendas           340                0
alcance de un equipo robado               datos + credencial   nada
telemetría por tienda                       1,4 GB/día       12 MB/día
avería detectada antes de la llamada           no        4 h antes, de media
```

La lección que esta clase traslada a la parte 13: los dos incidentes que costaron dinero no fueron de arquitectura distribuida: fueron una actualización desplegada a trescientas cuarenta tiendas a la vez sin forma de revertir, y un disco sin cifrar en un armario al que cualquiera podía llegar. Y el requisito que definió todo el diseño —cobrar sin conexión— se resolvió con la misma regla que la clase 147 dio para los contextos: que cada sitio sea el único escritor de sus datos, con un cupo asignado para lo único que no se podía partir.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-13-multicloud-hybrid-disaster-recovery/165-nube-hibrida-edge-y-conectividad-privada/lab.py
```

El laboratorio selecciona el motor de práctica hybrid 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-hibrida 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 híbrida con dependencia y modo degradado. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye arquitectura-hibrida 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
Al restablecerse la conexión, los sitios saturan el centro	Todos envían su cola a la vez, sin variación ni límite	Variación aleatoria al reconectar, limitación de caudal por sitio y cola acotada con criterio de descarte.
Una actualización deja sitios sin servicio y no hay forma de llegar a ellos	Se desplegó a todos a la vez, sin vigilante de arranque ni camino manual	Despliegue por grupos con tiempo entre etapas, reversión automática si la versión no arranca y un procedimiento manual sin red, probado.
Al reconciliar aparecen conflictos entre el sitio y el centro	Dos escritores para el mismo dato	Cada sitio es dueño de sus datos; lo que no se pueda partir, resuélvelo con cupos o exige conexión para esa operación.
Un equipo robado da acceso al sistema	Disco sin cifrar y credenciales permanentes en el sitio	Cifrado con la clave fuera del disco, arranque verificado, credenciales de vida corta revocables y permisos acotados a ese sitio.
Un sitio lleva días sin funcionar y nadie lo sabe	Ley 13: dejar de reportar no produce ningún error	Alerta por antigüedad del último informe por emplazamiento, distinguiendo falta de enlace de avería.
Se decide sacar cargas de la nube porque es cara	No se hizo la cuenta completa: hardware, energía, repuestos, personal y renovación	Compara coste total y con la vida útil del hardware; y escribe la fecha si el motivo es una inversión existente.

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 motivos justifican tener algo fuera de la nube pública y cuál define el diseño?
2. ¿Por qué la propiedad del dato por emplazamiento evita los conflictos al reconciliar?
3. ¿Qué tres problemas aparecen al recuperar la conexión?
4. ¿Qué impide que una actualización deje un sitio inaccesible?
5. ¿Por qué aquí el cifrado en reposo sí protege del escenario real?

Referencias

- AWS (2025). Outposts, Local Zones and edge services — espectro de opciones y su operabilidad. <https://docs.aws.amazon.com/whitepapers/latest/aws-outposts-high-availability-design/>
- Azure (2025). Azure Arc and hybrid management — gobierno de recursos fuera de la nube pública. <https://learn.microsoft.com/azure/azure-arc/overview>
- Google Cloud (2025). Distributed Cloud and edge patterns — proceso en el sitio y envío de resultados. <https://cloud.google.com/distributed-cloud>
- Eclipse Foundation (2025). Edge device update and rollback patterns — actualización con reversión automática. <https://projects.eclipse.org/>
- NIST (2025). Guide to industrial control system security — acceso físico, arranque verificado y credenciales en el sitio. <https://csrc.nist.gov/pubs/sp/800/82/r3/final>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Evaluación — 165 Nube híbrida, edge y conectividad privada

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-hibrida 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.

166 — Backup, RTO, RPO y patrones de disaster recovery

Parte: 13 — Multi-cloud, híbrido, migración y recuperación Nivel: experto · Horas estimadas: 4 Laboratorio: reliability · Estado: EXECUTABLECORE

Propósito

Prepararse para perder algo grande —una región, un proveedor, unos datos— con un plazo y una pérdida declarados y medidos. La clase separa tres cosas que se confunden a diario y protegen de amenazas distintas: alta disponibilidad, recuperación ante desastre y copia de seguridad. Insiste en que una réplica no es una copia, porque replica fielmente el borrado y la corrupción. Y demuestra que los objetivos declarados casi nunca se corresponden con nada medido, porque la cifra que se anuncia cubre solo la ejecución y el reloj real empieza mucho antes.

Resultados de aprendizaje

Al finalizar podrás:

1. Distinguir alta disponibilidad, recuperación ante desastre y copia de seguridad.
2. Declarar objetivos por escenario, no uno global.
3. Medir el plazo real de recuperación, con todos sus tramos.
4. Elegir el patrón por carga, según el impacto y no por costumbre.
5. Enumerar lo que no se replica y falla en una recuperación real.

Conceptos centrales

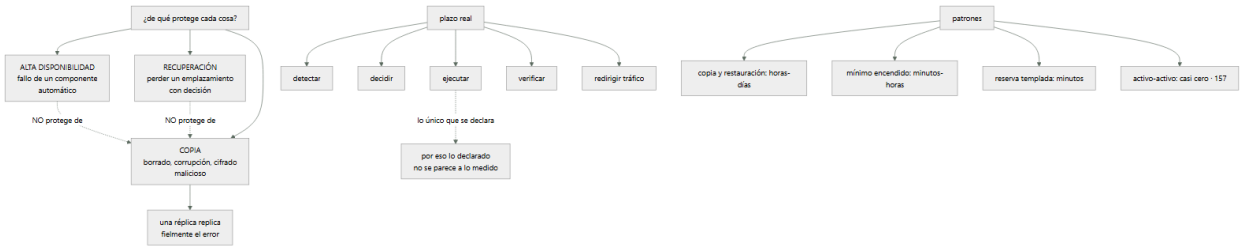
Concepto	Comprensión verificable
alta disponibilidad	Sobrevivir al fallo de un componente de forma automática, dentro del mismo sistema. No protege del borrado ni de la corrupción.
recuperación ante desastre	Sobrevivir a perder un emplazamiento entero, con un plan declarado y una decisión humana.
copia de seguridad	Punto en el tiempo al que volver. Es lo único que protege del borrado, de la corrupción y del cifrado malicioso.
plazo de recuperación	Tiempo desde que ocurre hasta que el servicio funciona. Incluye detectar, decidir, ejecutar, verificar y redirigir.
pérdida admisible	Cuántos datos se aceptan perder. Con copia asíncrona nunca es cero.
vuelta atrás	Regreso al emplazamiento original. Suele ser más difícil que la conmutación, porque el destino tiene ahora los datos nuevos.

Modelo mental

Multi-cloud es una decisión de negocio y riesgo; duplicar cada componente entre proveedores rara vez es la forma más confiable o económica de lograrla.

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

■ Flujo de razonamiento



Desarrollo

1. Tres cosas distintas

Se mencionan juntas y protegen de amenazas distintas:

```
ALTA DISPONIBILIDAD
  sobrevivir a que falle un componente: un nodo, una zona
  automática, sin decisión humana
  protege de   fallos de hardware, caída de una zona     clase 150
  NO protege de  borrado, corrupción, error de despliegue

RECUPERACIÓN ANTE DESASTRE
  sobrevivir a perder un emplazamiento entero
  con un plan, una decisión y un plazo declarado
  protege de   pérdida de una región o de un proveedor
  NO protege de  borrado ni corrupción

COPIA DE SEGURIDAD
  volver a un punto anterior en el tiempo
  protege de   borrado accidental, corrupción, error de
               despliegue, cifrado malicioso, y del error de
               un operador
```

Y la frase que hay que interiorizar:

```
UNA RÉPLICA NO ES UNA COPIA
  la réplica replica fielmente el DELETE que borró la tabla
  y lo hace en milisegundos                                clase 150
```

Y de ahí que las tres hagan falta y que ninguna sustituya a otra:

```
un sistema con réplica en tres zonas y sin copias
→ está protegido de que se caiga una zona
→ y no lo está de que alguien ejecute una migración mal escrita
```

Y la lista de amenazas contra las que solo protege la copia:

```
borrado accidental de datos o de recursos
corrupción por un fallo de la aplicación
una migración de esquema que destruye información       clase 102
cifrado malicioso por un atacante
borrado deliberado por alguien con permisos             clase 134
y un error de configuración que vacía un almacén        clase 139
```

Y una consecuencia sobre las copias que se deriva de las dos últimas:

```
si quien administra puede borrar las copias, no protegen del caso
en que su credencial esté comprometida
→ inmutabilidad con retención, y cuenta separada         clases 112, 133
```

2. Objetivos por escenario

Declarar «nuestro plazo de recuperación es de cuatro horas» sin decir de qué no significa nada. Los objetivos se declaran por escenario:

```
ESCENARIO                       plazo        pérdida admisible
falla un nodo                   segundos     cero        automático
falla una zona                  minutos      cero        automático
se pierde una región            horas        minutos     con decisión
se pierde el proveedor          horas-días   minutos     con decisión
corrupción de datos             horas        depende del punto elegido
borrado accidental              minutos-horas  depende
cifrado malicioso               días         desde la última copia
```

```
                                                        inmutable
```

Y dos observaciones sobre la tabla:

```
los escenarios de arriba se resuelven con alta disponibilidad y
  no necesitan plan
los de abajo necesitan copia, y su plazo lo domina el tamaño
  de los datos                                          clase 161
```

Y los objetivos se declaran por carga, no para el sistema entero:

```
flujo de compra          plazo corto, pérdida cero
informes y análisis      plazo de días, pérdida de horas
herramientas internas    plazo de días
→ y eso permite gastar donde importa                    clase 143
```

Y la pregunta que fija cada número, que es la de la clase 145:

```
¿qué pasa si tardamos el doble?
¿qué pasa si perdemos una hora de datos en vez de un minuto?
→ con la respuesta en euros o en consecuencias, el número deja de
  ser una aspiración
```

Y una cifra que casi nunca se calcula y que cambia las decisiones:

```
COSTE DE LA PREPARACIÓN frente a COSTE DEL SUCESO × PROBABILIDAD

reserva templada entre proveedores          ~11.400 €/mes
pérdida estimada de un día de caída          ~90.000 €
probabilidad anual de perder el proveedor    muy baja
probabilidad anual de perder una región      moderada
→ y por eso la respuesta suele ser una segunda REGIÓN y no un
  segundo proveedor                                     clase 157
```

3. Medir el plazo de verdad

Aquí está el motivo por el que lo declarado y lo medido no se parecen: el número que se anuncia cubre un tramo y el reloj corre desde antes.

```
el reloj empieza cuando ocurre, no cuando alguien lo mira

1. DETECTAR            ¿cuánto tarda alguien en saberlo?      parte 10
2. DECIDIR             ¿quién decide conmutar? ¿está localizable?
                       ¿hay criterio escrito o se delibera?
3. EJECUTAR            ← lo único que suele estar medido
4. VERIFICAR           ¿funciona de verdad? ¿con qué comprobación?
5. REDIRIGIR TRÁFICO   nombres, cachés de resolución, clientes
                       con conexiones abiertas               clase 160
```

Y dos tramos que se olvidan y que dominan cuando hay datos:

```
TRANSFERIR LOS DATOS   si hay que traerlos, cuesta tiempo y dinero
                                                          clase 161
CALENTAR               cachés vacías, conexiones frías, código sin
                       optimizar: el sistema arranca degradado
                                                          clase 111
```

Y la forma de medirlo es una sola:

```
EJECUTAR EL PLAN, con cronómetro, en un ensayo real
→ cualquier otra cifra es una estimación de alguien
```

Y lo que aparece la primera vez, sin falta:

```
el procedimiento está desactualizado
falta un permiso                                        clase 131
el documento vive en el sistema que se ha caído
la cuota de la región de destino no da para toda la carga  clase 129
faltan los certificados, o han caducado                 clase 136
los secretos no están en el destino                      clase 137
el proveedor externo solo acepta llamadas desde las direcciones
  de origen                                              clase 135
y nadie sabe quién tiene autoridad para decidir
```

Y el criterio de decisión, que es la parte organizativa y la que más tarda:

```
escrito de antemano
  «si el indicador está por debajo de X durante Y minutos
```

```
   y no hay perspectiva de recuperación en Z, se conmuta»
y con nombre de quién decide, y suplente
→ sin eso, la decisión tarda más que la ejecución
```

Y la asimetría que conviene aceptar:

```
conmutar demasiado pronto        se pierde algo de datos y se recupera
conmutar demasiado tarde         se acumula la caída
→ y como conmutar tiene coste, la tentación es esperar
→ por eso el criterio se fija antes, no durante
```

4. Patrones, lo que no se replica y la vuelta

Los cuatro patrones, con su plazo realista y su coste:

```
COPIA Y RESTAURACIÓN
  no hay nada encendido; se crea todo al ocurrir
  plazo   horas o días, dominado por el tamaño de los datos
  coste   el de guardar las copias
  encaja  cargas que toleran estar caídas un día

MÍNIMO ENCENDIDO
  los datos se replican y la infraestructura está declarada,
  con lo mínimo encendido
  plazo   minutos a horas: hay que crear y escalar
  coste   bajo: solo los datos y unos pocos recursos
  encaja  la mayoría de los casos

RESERVA TEMPLADA
  una copia reducida corriendo, con datos al día
  plazo   minutos: escalar y redirigir
  coste   medio-alto

ACTIVO-ACTIVO
  sirve desde los dos                                  clase 157
  plazo   casi cero
  coste   alto, y todo lo de los niveles 4 y 5
```

Y el criterio de elección, por carga:

```
se elige el patrón MÁS BARATO que cumpla el objetivo declarado
y el objetivo sale del impacto, no de la ambición
```

Lo que no se replica y falla en una recuperación real. Esta lista es lo más útil de la clase:

```
nombres y su resolución                              clase 160
certificados y su cadena de confianza                clase 136
secretos y credenciales del destino                  clase 137
identidades, roles y permisos                        clase 159
cuotas y límites de la región de destino             clase 129
registro de imágenes accesible desde allí            clase 138
listas de direcciones autorizadas en proveedores externos
configuración de la puerta de entrada y sus límites  clase 118
reglas de red y de salida                            clase 135
programación de trabajos periódicos
y el propio procedimiento, si vive en lo que se ha caído
```

Y las dos últimas causan más problemas de lo que parece:

```
los trabajos periódicos no se activan solos en el destino
  → o no se ejecutan, o se ejecutan DOS veces si el origen revive
y un plan almacenado en el sistema caído no existe cuando hace falta
```

Las copias, con lo mínimo exigible:

```
en otra cuenta y con credenciales distintas           clase 133
inmutables, con retención                             clase 112
varias generaciones, y una fuera del proveedor si el escenario
  incluye perderlo
cifradas, con la clave disponible en el destino       clase 136
y RESTAURADAS periódicamente, con cronómetro
```

Y la última es la única que cuenta: una copia no restaurada no es una copia, que este programa lleva repitiendo desde la clase 088.

La vuelta atrás, que es la mitad olvidada:

```
tras conmutar, el destino tiene los datos NUEVOS
→ volver significa replicar en sentido contrario
→ y decidir qué se hace con lo que el origen tenía y no llegó a copiar

y si el origen revive solo, hay riesgo de dos escritores    ley 21
→ hace falta un mecanismo que impida que el origen vuelva a aceptar
  escrituras sin decisión explícita                       clase 150
```

Y la lista de comprobación de la clase:

- ■ están separadas alta disponibilidad, recuperación y copia
- ■ hay copias, y no solo réplicas
- ■ las copias son inmutables, en otra cuenta y con credenciales distintas
- ■ los objetivos están declarados por escenario y por carga
- ■ cada número se justifica con la consecuencia de incumplirlo
- ■ el plazo medido incluye detectar, decidir, verificar y redirigir
- ■ está escrito el criterio de decisión y quién decide, con suplente
- ■ el patrón elegido es el más barato que cumple el objetivo
- ■ está revisada la lista de lo que no se replica
- ■ el procedimiento no vive en el sistema que puede caerse
- ■ las cuotas del destino dan para toda la carga
- ■ se restaura periódicamente, con cronómetro
- ■ está previsto el regreso y qué impide dos escritores

Y el cierre que enlaza con la clase siguiente: casi todo lo anterior supone que las cargas ya están donde tienen que estar. Cuando hay que llevarlas de un sitio a otro —a la nube, entre nubes o de vuelta— hay siete formas de hacerlo, con costes muy distintos, y es la materia de la clase 167.

Ejemplo trabajado

CloudShop tiene un documento de continuidad que declara un plazo de recuperación de cuatro horas para el flujo de compra. Nunca se ha ejecutado. El ejercicio consiste en ejecutarlo con un cronómetro.

Lo declarado.

```
plazo de recuperación declarado                          4 h
pérdida admisible declarada                          15 min
escenario cubierto                  «pérdida de la región principal»
patrón supuesto                     «mínimo encendido»
última revisión del documento                        hace 19 meses
veces que se ha ejecutado                              0
```

El primer ensayo, con cronómetro.

```
09:00  se simula la pérdida de la región principal
09:00  → el reloj empieza aquí

09:00  DETECTAR
       la sonda externa avisa a los                      90 s
       ✓ funcionó (clase 162)

09:02  DECIDIR
       criterio escrito                                  no había
       personas que debían autorizar                     2, no localizadas
       tiempo hasta la decisión                          1 h 40

10:42  EJECUTAR
       el documento estaba en un espacio de la región caída
         → se recuperó de una copia local de alguien     22 min
       faltaban permisos para crear en la región de destino 35 min
       la cuota de instancias del destino era de 40; hacían falta 120
         → petición al proveedor                          2 h 10
       los certificados no estaban en el destino         18 min
       los secretos tampoco                              41 min
       restaurar la base desde la copia                  3 h 05
         (4,1 TB; el tiempo de transferencia lo dominaba)

16:31  VERIFICAR
       no había comprobación definida; se improvisó      38 min

17:09  REDIRIGIR
       el tiempo de vida de los registros de nombres era de 3.600 s
         → clientes llegando al origen                    1 h 12
```

```
        el proveedor de pago rechazaba las llamadas: solo
          acepta direcciones autorizadas y las nuevas no estaban
                                                          1 h 30

19:51  el flujo de compra funciona

PLAZO REAL                                            10 h 51
PLAZO DECLARADO                                            4 h
factor                                                   ×2,7
```

Y la pérdida de datos:

```
retardo de replicación en el momento del corte              41 s
pérdida real                                          ~1.100 pedidos
pérdida declarada admisible                           15 min
→ dentro del objetivo, por poco                              ✓
```

El desglose por tramos, que es lo que orientó el trabajo.

```
detectar                                    90 s        0,2 %
decidir                                  1 h 40       15,4 %
ejecutar                                 6 h 21       58,5 %
  de los cuales, cuota del destino       2 h 10
  de los cuales, transferencia de datos  3 h 05
verificar                                   38 min       5,8 %
redirigir                                2 h 42       24,9 %
```

El 41 % del tiempo no era ejecutar nada: era decidir, verificar y redirigir, que es exactamente lo que el número declarado no cubría.

Las correcciones, por orden de rendimiento.

```
1. CRITERIO DE DECISIÓN ESCRITO
   «si el indicador del flujo de compra está por debajo del 50 %
    durante 10 minutos y el proveedor no da plazo de recuperación
    en 30, se conmuta»
   quién decide: la persona de guardia; suplente nombrado
   ahorro                                              1 h 40 → 4 min

2. CUOTAS EN EL DESTINO, SOLICITADAS DE ANTEMANO
   cuota de instancias                          40 → 150
   cuota de concurrencia, de direcciones y de bases  igual
   ahorro                                              2 h 10 → 0
   coste                                               0 €

3. DATOS: DE RESTAURAR A REPLICAR
   copia continua a la región de destino                clase 161
   coste                                              ~140 €/mes
   ahorro                                              3 h 05 → 6 min

4. REDIRECCIÓN
   tiempo de vida de los registros de nombres      3.600 s → 60 s
   direcciones del destino autorizadas en el proveedor de pago,
     de antemano
   ahorro                                              2 h 42 → 9 min

5. CERTIFICADOS Y SECRETOS EN EL DESTINO
   emitidos y replicados de antemano                   clases 136, 137
   ahorro                                              59 min → 0

6. VERIFICACIÓN DEFINIDA
   una comprobación automática que ejecuta el recorrido de compra
   ahorro                                              38 min → 3 min

7. EL PROCEDIMIENTO FUERA DE LA REGIÓN
   en un repositorio replicado y con copia impresa en dos oficinas
   ahorro                                              22 min → 0
```

El segundo ensayo, tres meses después.

```
detectar                                  80 s
decidir                                   4 min
ejecutar                                 14 min
verificar                                 3 min
redirigir                                 9 min
```

```
                                          ■■■■■■
PLAZO REAL                                31 min

plazo declarado, actualizado              45 min
pérdida real                              22 s de datos
```

De diez horas y cincuenta y un minutos a treinta y uno, sin cambiar de patrón: sigue siendo mínimo encendido. Lo que cambió fue todo lo que rodea a la ejecución.

La copia que no era copia.

Durante la preparación se descubrió algo peor que el plazo:

```
«tenemos réplica en tres zonas»                          sí
«y copias de seguridad»                                  sí, diarias
¿dónde?              en la misma cuenta y el mismo proveedor
¿inmutables?         no
¿quién puede borrarlas?   el mismo rol que administra la base
¿restauradas alguna vez?  una, hace 14 meses
```

Y el ensayo del escenario de borrado, que es el que las réplicas no cubren:

```
se simuló una migración que borra una tabla
  las 3 réplicas la borraron en 40 ms                    ✓ (esperado)
  restauración desde copia                               4 h 20
  datos perdidos entre la copia y el borrado             17 h
```

```
copias                              misma cuenta      cuenta separada
inmutabilidad                            no        sí, 35 días
quien puede borrarlas               el administrador   nadie, dentro
                                    de la base         de la retención
frecuencia                          diaria          continua + diaria
pérdida en el escenario de borrado       17 h            5 min
restauración probada                hace 14 meses     trimestral
tiempo de restauración medido            4 h 20          38 min
```

Los objetivos, redeclarados por escenario y por carga.

```
flujo de compra
  pérdida de zona                    automático     0
  pérdida de región                    45 min      1 min
  corrupción o borrado                  1 h        5 min
catálogo
  pérdida de región                     4 h        1 h
informes y análisis
  pérdida de región                    24 h        24 h
herramientas internas
  pérdida de región                     3 días     1 día
```

Y el coste de esa diferenciación:

```
coste si todo tuviera el objetivo del flujo de compra   ~4.900 €/mes
coste con objetivos por carga                             ~610 €/mes
```

La vuelta atrás, ensayada aparte.

```
primer ensayo de regreso                            falló
  la región original revivió y empezó a aceptar escrituras
  → dos escritores durante 4 minutos                    ley 21
  → 61 pedidos escritos en el sitio equivocado

corrección
  al conmutar, el origen se marca como no autorizado a escribir
  y volver exige una decisión explícita y replicar en sentido inverso

segundo ensayo de regreso                           correcto
  duración                                           52 min
  pedidos escritos en el sitio equivocado                 0
```

A los seis meses.

```
plazo declarado                          4 h           45 min
plazo medido                            10 h 51        31 min
relación entre declarado y medido        ×2,7          ×0,7
ensayos ejecutados                         0             4
criterio de decisión escrito              no            sí
cuotas del destino preparadas             no            sí
```

```
certificados y secretos en el destino           no             sí
procedimiento fuera de la región caída          no             sí
copias inmutables en cuenta separada            no             sí
pérdida en escenario de borrado                17 h            5 min
restauración probada                       hace 14 meses    trimestral
vuelta atrás ensayada                           no             sí
objetivos por carga                              1              4
```

La lección que esta clase traslada a la parte 13: el plazo declarado era de cuatro horas y el real de casi once, y el 41 % de ese tiempo no fue ejecutar nada: fue decidir sin criterio escrito, verificar sin comprobación definida y redirigir con un tiempo de vida de nombres de una hora. Y el hallazgo más grave no tenía que ver con el plazo: las copias vivían en la misma cuenta que la base y las podía borrar el mismo rol, de modo que el único escenario del que una réplica no protege —el borrado— era también el peor cubierto.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-13-multicloud-hybrid-disaster-recovery/166-backup-rto-rpo-y-patrones-de-disaster-recovery/lab.py
```

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

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

Evidencia esperada

El artefacto principal es un escenario de fallo con objetivo y recuperación medida. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye plan-dr 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 réplicas en varias zonas y un borrado destruye los datos igualmente	Una réplica replica fielmente el borrado; no es una copia	Copias con punto en el tiempo, inmutables, en otra cuenta y con credenciales distintas.
El plazo real triplica al declarado	El número declarado solo cubría la ejecución; el reloj corre desde que ocurre	Mide con cronómetro los cinco tramos: detectar, decidir, ejecutar, verificar y redirigir.
La decisión de conmutar tarda más que la conmutación	No hay criterio escrito ni quién decide	Escribe el criterio con umbrales y plazos, nombra a quien decide y a su suplente.
Al conmutar no hay capacidad en la región de destino	Las cuotas del destino son las de una región que no se usaba	Solicita de antemano las cuotas necesarias y verifícalas en cada ensayo.
El sistema se levanta en el destino y no funciona	Falta lo que no se replica: certificados, secretos, permisos, listas de direcciones autorizadas y trabajos periódicos	Revisa la lista completa y prepara cada elemento antes de necesitarlo.
Al volver al sitio original aparecen datos escritos en dos sitios	El origen revivió y aceptó escrituras	Marca el origen como no autorizado a escribir al conmutar y exige decisión explícita para volver, replicando en sentido inverso.

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 cada una: alta disponibilidad, recuperación ante desastre y copia?
2. ¿Por qué una réplica no es una copia?
3. ¿Qué cinco tramos componen el plazo real de recuperación y cuál se suele declarar?
4. ¿Qué elementos no se replican y hacen fallar una recuperación real?
5. ¿Por qué la vuelta atrás suele ser más difícil que la conmutación?

Referencias

- AWS (2025). Disaster recovery of workloads: backup, pilot light, warm standby, active/active — patrones y sus plazos. <https://docs.aws.amazon.com/whitepapers/latest/disaster-recovery-workloads-on-aws/disaster-recovery-options-in-the-cloud.html>
- Google Cloud (2025). Disaster recovery planning guide — objetivos por escenario y verificación. <https://cloud.google.com/architecture/dr-scenarios-planning-guide>
- Azure (2025). Business continuity and backup immutability — copias inmutables y separación de credenciales. <https://learn.microsoft.com/azure/backup/backup-azure-immutable-vault-concept>
- NIST (2010). SP 800-34: contingency planning — definición de objetivos y pruebas del plan. <https://csrc.nist.gov/pubs/sp/800/34/r1/upd1/final>
- Google SRE (2025). Data integrity: backups versus replication — por qué la replicación no protege del borrado. <https://sre.google/sre-book/data-integrity/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 13 en PDF · Manual integral

Evaluación — 166 Backup, RTO, RPO y patrones de disaster recovery

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 plan-dr 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.

167 — Las 7R de migración y oleadas

Parte: 13 — Multi-cloud, híbrido, migración y recuperación Nivel: experto · Horas estimadas: 4 Laboratorio: migration · Estado: EXECUTABLECORE

Propósito

Llevar cargas de un sitio a otro —a la nube, entre nubes o de vuelta— eligiendo por carga y no una estrategia para todas. La clase ordena las siete formas de hacerlo por coste, riesgo y beneficio, y defiende que la primera y la más rentable es retirar lo que ya no hace falta, que es también la que más se salta. Después trata las dos cosas que deciden si el proyecto sale bien: el descubrimiento, porque no se puede mover lo que no se sabe que existe, y las oleadas, que deben ser lo bastante pequeñas como para poder deshacerlas.

Resultados de aprendizaje

Al finalizar podrás:

1. Elegir una de las siete formas por carga, con criterio.
2. Descubrir lo que existe de verdad, incluidas las dependencias no documentadas.
3. Secuenciar en oleadas pequeñas y reversibles.
4. Convivir durante la transición sin duplicar escritores.
5. Validar que la migración funcionó, y medir el programa por lo correcto.

Conceptos centrales

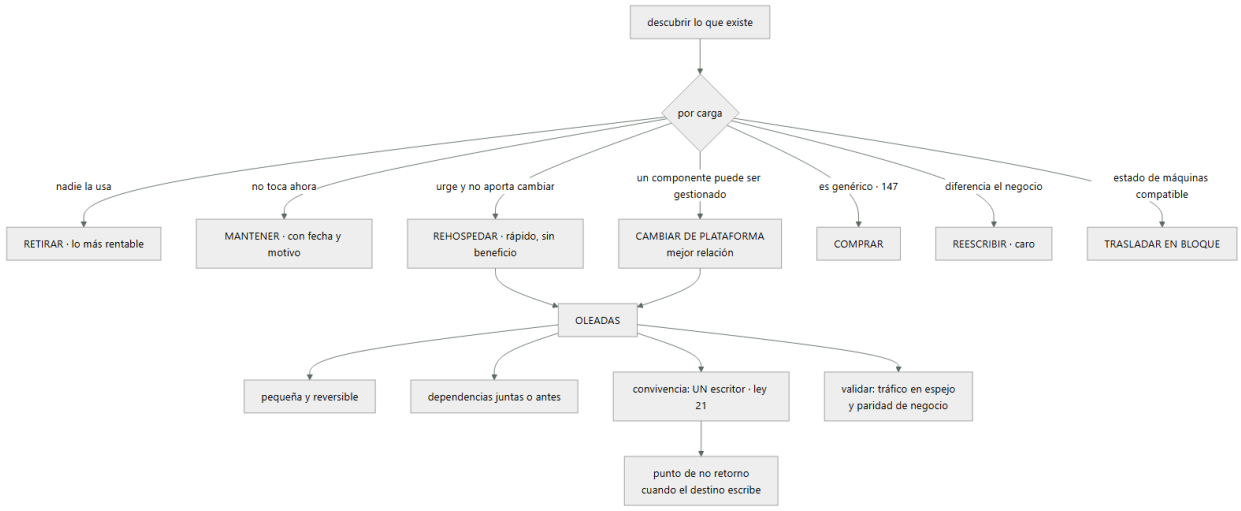
Concepto	Comprensión verificable
retirar	Apagar lo que ya no se usa. Es la opción más barata y la de mayor rendimiento, y se descubre solo al inventariar.
rehospedar	Mover tal cual, sin cambiar nada. Rápido y barato, y traslada los problemas junto con la carga.
cambiar de plataforma	Sustituir un componente por su equivalente gestionado sin tocar la aplicación. Suele ser la mejor relación entre coste y beneficio.
oleada	Conjunto de cargas que se migran juntas. Debe ser pequeña, con dependencias resueltas y reversible.
convivencia	Periodo en que origen y destino funcionan a la vez. Su problema central es que solo uno puede escribir.
punto de no retorno	Momento en que ya se han escrito datos en el destino y volver deja de ser gratis.

Modelo mental

Multi-cloud es una decisión de negocio y riesgo; duplicar cada componente entre proveedores rara vez es la forma más confiable o económica de lograrla.

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. Siete formas, y la que se salta

```
RETIRAR
  apagarlo: ya no se usa
  coste     casi nulo    beneficio  el mayor de todos
  → y siempre hay más de lo que nadie cree: 10-25 % del inventario

MANTENER
  dejarlo donde está, por ahora
  → legítimo, y con FECHA y motivo escritos, o se queda para siempre

REHOSPEDAR
  mover tal cual, sin cambiar nada
  coste     bajo    riesgo  bajo    beneficio  ninguno por sí solo
  → traslada los problemas junto con la carga
  → tiene sentido cuando hay una fecha límite: un contrato que vence,
    un centro que cierra
  → y entonces el beneficio llega DESPUÉS, si se sigue trabajando

CAMBIAR DE PLATAFORMA
  sustituir un componente por su equivalente gestionado
  la base propia por una gestionada; el servidor de aplicaciones
  por contenedores
  coste     medio   beneficio  alto
  → suele ser la mejor relación de las siete

COMPRAR
  sustituirlo por un producto
  → para lo genérico, según la clasificación de la clase 147

REESCRIBIR
  rehacerlo aprovechando lo que la nube ofrece
  coste     el mayor    riesgo  el mayor
  → solo para lo que diferencia al negocio

TRASLADAR EN BLOQUE
  mover un conjunto de máquinas a un destino compatible, sin convertir
  → rápido cuando existe esa compatibilidad, y conserva las
    limitaciones del origen
```

Y la observación que ordena la clase:

```
RETIRAR es la primera opción y la que casi nadie considera
→ porque exige saber qué se usa, y eso obliga a inventariar
→ y porque nadie se apunta el mérito de apagar cosas
```

Y el error más común, que es elegir una sola forma para todo:

```
«vamos a rehospedarlo todo y luego ya veremos»
  → se paga la migración y no se obtiene ningún beneficio
```

```
  → y el «luego» no llega

«vamos a reescribirlo todo»
  → el plazo se multiplica y el proyecto se cancela a mitad
```

Y el criterio por carga:

```
¿la usa alguien?             no → retirar
¿diferencia al negocio?      no → comprar o cambiar de plataforma
¿hay fecha límite dura?      sí → rehospedar y mejorar después
¿qué componente es el que duele?  → cambiar ese, no todo
```

2. Descubrir, que decide el proyecto

No se puede migrar lo que no se sabe que existe. Y el inventario documentado nunca coincide con la realidad; este programa lo ha medido tres veces:

```
dependencias documentadas 23, observadas 41              clase 124
conexiones documentadas 23, observadas 58                clase 135
cuentas documentadas 14, existentes 23                   clase 139
```

Lo que hay que descubrir, y con qué:

```
QUÉ EXISTE
  de la facturación, no del inventario                   clase 139
  y de lo que consume red, aunque no esté en ninguna lista

QUIÉN LO USA
  tráfico observado, registros de acceso, y preguntar
  → lo que no recibe tráfico en 90 días es candidato a retirar

DE QUÉ DEPENDE
  conexiones observadas, no el diagrama                  clase 135
  incluidas las salientes hacia terceros

QUÉ DATOS TIENE
  volumen, crecimiento y clasificación                   clase 141
  → el volumen decide el plazo de la migración

QUÉ LICENCIAS Y CONTRATOS LLEVA
  y si son válidas en el destino                         ← se olvida

QUIÉN ES SU DUEÑO
  y si no lo tiene, ese es el primer problema            ley 20
```

Y las dos preguntas que más ahorran, aplicadas a cada elemento:

```
¿qué pasa si lo apagamos un día?
  → la forma más rápida de saber si alguien lo usa
  → y se puede hacer de verdad, con aviso, en un entorno controlado

¿quién nos llamaría?
  → si nadie sabe responder, probablemente nadie
```

Y la técnica que resuelve las dependencias ocultas sin adivinar:

```
registrar durante semanas quién habla con quién
y quién llama a quién desde fuera
→ y solo entonces agrupar en oleadas
```

Y un aviso sobre el calendario: el descubrimiento tarda más de lo que nadie planifica, y recortarlo es lo que produce las sorpresas del apartado cuarto.

3. Oleadas y convivencia

Las oleadas ordenan el trabajo, y sus tres criterios:

```
1. DEPENDENCIAS JUNTAS O ANTES
   lo que se llama mucho entre sí, se mueve junto
   → si no, cada llamada cruza la frontera y paga latencia y salida
                                                  clases 160, 161

2. LO SENCILLO PRIMERO
   la primera oleada sirve para construir la maquinaria:
   canalización, red, identidad, observabilidad, procedimiento
   → y para que el equipo aprenda con algo que no duele
```

```
3. LO BASTANTE PEQUEÑA COMO PARA DESHACERLA
   si una oleada no se puede revertir, no es una oleada:
   es una migración completa con otro nombre
```

Y el antipatrón correspondiente:

```
MIGRACIÓN DE UNA SOLA VEZ
  todo un fin de semana, sin vuelta atrás
  → y si algo sale mal, no hay decisión que tomar: hay que seguir
  → se elige por comodidad de calendario, no por riesgo
```

La convivencia, que es el problema técnico central:

```
durante la transición, origen y destino existen a la vez
y la pregunta es siempre la misma:

¿QUIÉN ESCRIBE LOS DATOS?                                  ley 21
```

Y las tres respuestas, con lo que implican:

```
MOVER LOS DATOS PRIMERO, y luego la carga
  el destino es el único escritor desde el principio
  + sin conflictos
  – hay una ventana de parada mientras se mueven

MOVER LA CARGA PRIMERO, con los datos en el origen
  el destino escribe en la base del origen, por red
  + la parada es mínima
  – latencia y salida en cada operación                  clase 161
  – y solo sirve si la latencia lo tolera

SINCRONIZAR EN LOS DOS SENTIDOS
  – conflictos garantizados                             clase 149
  → se evita salvo que se pueda partir por cliente o por región
```

Y el patrón que permite migrar por partes sin parar nada:

```
SUSTITUCIÓN PROGRESIVA
  una fachada delante decide qué peticiones van al origen y cuáles
  al destino, por funcionalidad o por porcentaje
  se mueve una capacidad, se comprueba, y se pasa a la siguiente
  → es el escalonado de la clase 102 aplicado a una migración

y su requisito: los datos de esa capacidad tienen que estar
accesibles desde el lado que la sirve
→ que es de nuevo la pregunta de quién escribe
```

Y el punto de no retorno, que es el de la clase 102 a escala de migración:

```
mientras el destino solo LEA, volver es gratis
en cuanto el destino ESCRIBE, volver significa traerse esos datos
→ y hay que decidir por adelantado en qué momento se cruza
→ y qué se hace si hay que volver después de cruzarlo
```

4. Lo que siempre sale mal, y cómo se valida

La lista de sorpresas, que se repite en casi todas las migraciones:

```
LICENCIAS
  válidas en el hardware antiguo y no en el destino, o con otro
  modelo de cómputo
  → y el proveedor del software se entera y factura

RENDIMIENTO DISTINTO
  el procesador, el disco y la red no se comportan igual
  → una carga que iba justa, va peor
  → y hay que medir antes, no después                    clase 129

DEPENDENCIAS QUE APARECEN AL CORTAR
  un trabajo mensual, un informe trimestral, un sistema que llamaba
  una vez al día
  → por eso el descubrimiento debe durar más de un mes

VOLUMEN Y TIEMPO DE LOS DATOS
  transferir lo que hay tarda lo que tarda               clase 161
  → y esa cifra decide la ventana, no al revés
```

```
DIRECCIONAMIENTO E IDENTIDAD
  rangos que solapan, nombres que cambian, permisos que no existen
                                                    clases 159, 160

DIRECCIONES AUTORIZADAS EN TERCEROS
  el proveedor de pago solo acepta llamadas desde las direcciones
  del origen                                        clase 166

LA VENTANA DE CORTE SE QUEDA CORTA
  y hay que decidir a las 4 de la madrugada si se sigue o se vuelve
  → por eso el criterio de vuelta atrás se escribe ANTES
```

La validación, que decide si la migración funcionó:

```
TRÁFICO EN ESPEJO
  se envían las mismas peticiones a origen y destino y se comparan
  las respuestas                                    clase 153
  → detecta diferencias antes de mover a nadie

PARIDAD DE NEGOCIO
  pedidos por hora, importes, tasas de conversión
  → si el sistema técnico va bien y el negocio baja, algo falla

COMPARACIÓN DE DATOS
  recuentos y sumas de control por tabla, antes y después

Y EL PLAZO DE OBSERVACIÓN
  no se declara terminada una oleada el mismo día
  → los procesos mensuales y trimestrales no han corrido todavía
```

La última evita el error más común de todos: dar por buena una migración antes de que se ejecute el ciclo completo del negocio.

Cómo se mide el programa, que es donde se engaña más gente:

```
mal   «cargas migradas»
      → cuenta igual una que se retiró que una que se rehospedó sin
        beneficio
      → y no dice si el sistema está mejor

bien  cargas retiradas
      cargas migradas, por forma elegida
      coste antes y después, por unidad de negocio        clase 142
      latencia y disponibilidad antes y después
      incidentes causados por la migración
      y cargas que siguen en «mantener» con su fecha vencida
```

Y la última fila es la que evita el final habitual de estos programas: un resto que nadie migra y que mantiene vivo el origen entero, con su coste y su equipo.

Y la lista de comprobación de la clase:

- el inventario sale de la facturación y del tráfico, no de una lista
- se ha preguntado por cada elemento si alguien lo usa
- las dependencias se han observado durante semanas, no supuesto
- cada carga tiene una forma elegida, con motivo
- lo que se mantiene tiene fecha y dueño
- las oleadas son pequeñas y reversibles
- la primera oleada es sencilla y sirve para montar la maquinaria
- está decidido quién escribe los datos durante la convivencia
- está escrito cuándo se cruza el punto de no retorno
- hay criterio de vuelta atrás escrito antes del corte
- se ha comprobado licencias, rendimiento y direcciones autorizadas
- hay validación con tráfico en espejo y paridad de negocio
- una oleada no se cierra hasta pasar un ciclo completo de negocio
- el programa se mide por retiradas, coste y calidad, no por cargas movidas

Y el cierre que enlaza con la clase siguiente: con todo lo de esta parte —motivos, portabilidad, identidad, red, datos, operación, infraestructura, flota, borde, recuperación y migración—, queda montar la continuidad de verdad entre dos proveedores y comprobarla. Y con ella, calificar las cinco predicciones de la clase 156. Es la materia de la clase 168.

Ejemplo trabajado

CloudShop migra el patrimonio de la empresa adquirida: cuarenta y una cargas en un centro de datos propio con contrato que vence en once meses. El ejercicio empieza inventariando y termina con doce cargas apagadas y una oleada revertida.

El descubrimiento, y lo que no estaba en la lista.

```
cargas según el inventario recibido                              29
cargas encontradas en la facturación y en el tráfico             41
diferencia                                                       12

de las 41
  con dueño identificable                                        22
  sin tráfico en 90 días                                         14
  con dependencias no documentadas                               18
  con licencias de terceros                                       9
  con datos personales                                            6
```

Y la pregunta del apartado segundo, aplicada a las catorce sin tráfico:

```
«¿qué pasa si lo apagamos un día?»
  se apagaron 14, con aviso, un martes
  llamadas recibidas                                              2
    → una era un informe trimestral que corría en un mes distinto
    → la otra, una integración con un proveedor externo
  las otras 12 no las echó de menos nadie
```

```
RETIRADAS                                                        12
coste mensual que desaparece                                1.840 €
esfuerzo de migración evitado                              ~9 semanas
```

Doce de cuarenta y una, y ninguna estaba marcada como candidata en el inventario recibido.

La forma elegida por carga.

```
RETIRAR                                                          12
MANTENER                                                          3
  → un sistema de un cliente cuyo contrato termina en 8 meses
  → con fecha escrita y dueño
REHOSPEDAR                                                       11
  → contrato del centro de datos vence: hay fecha límite dura
  → y se anota qué se hará después con cada una
CAMBIAR DE PLATAFORMA                                             9
  → 6 bases propias a bases gestionadas
  → 3 servidores de aplicaciones a contenedores
COMPRAR                                                           4
  → correo, mensajería, facturación fiscal y firma electrónica
REESCRIBIR                                                        2
  → el motor de tarifas, que sí diferencia                 clase 147
TRASLADAR EN BLOQUE                                               0
```

Y la decisión de rehospedar once, con su condición:

```
se rehospedan porque hay fecha límite
y cada una lleva escrito qué se hará después y cuándo
  6 pasarán a plataforma gestionada en los 12 meses siguientes
  3 se retirarán cuando su función se absorba
  2 quedarán como están, y eso también está escrito
```

Las oleadas.

```
oleada 1   2 cargas sencillas, sin datos y sin dependencias
           objetivo real: montar canalización, red, identidad,
           observabilidad y procedimiento
           duración                                      3 semanas
           incidencias                                   11
           → todas de maquinaria, ninguna de las cargas

oleada 2   6 cargas del mismo grupo de dependencias
oleada 3   9 cargas, incluidas 4 con base de datos
oleada 4   8 cargas, las más grandes
oleada 5   2 reescrituras, con calendario propio
```

Y el criterio de agrupación salió del registro de conexiones:

```
conexiones observadas en 6 semanas                              88
grupos que se llaman mucho entre sí                              5
cargas que había que mover juntas por latencia                  14
```

La oleada que hubo que revertir.

```
oleada 3, corte del sábado
02:00  se mueve el tráfico de 9 cargas al destino
02:40  una de ellas, un sistema de reservas, empieza a dar errores
03:10  causa: latencia de 41 ms hacia la base, que seguía en el origen
       la aplicación abría una conexión por consulta
       en el centro de datos eran 0,4 ms y no importaba
03:20  decisión de revertir, según el criterio escrito
03:50  revertido

tiempo total                                              1 h 50
datos escritos en el destino                                   0
  → todavía no se había cruzado el punto de no retorno
```

Y lo que permitió que la reversión fuera limpia:

```
el criterio estaba escrito: «si algún indicador está por debajo
  del 95 % 20 minutos después del corte, se revierte»
el destino aún no escribía datos propios
y la fachada podía devolver el tráfico al origen en minutos
```

Y la corrección antes de repetirla:

```
conexiones por consulta                      una          agrupador  clase 109
consultas por operación                       41              3
latencia tolerada                           0,4 ms          41 ms
y sobre todo   se movió la base ANTES que la aplicación
oleada 3 repetida, dos semanas después                    correcta
```

Las sorpresas del apartado cuarto, contadas.

```
licencias no válidas en el destino                              3
  coste adicional no previsto                          14.000 €/año
  → una se sustituyó por producto comprado, dos se renegociaron

rendimiento distinto                                            4 cargas
  → 2 necesitaron más recursos, 2 fueron más rápidas

dependencias que aparecieron al cortar                          5
  → 3 trabajos mensuales, 1 informe trimestral,
    1 integración externa que llamaba una vez al día

direcciones autorizadas en terceros                             2
  → el proveedor de pago y una pasarela bancaria
  → detectadas en el ensayo, no en el corte

ventana de corte insuficiente                                   1
  → la oleada 4: la transferencia de datos tardaba 9 h y la ventana
    era de 6
  → se resolvió con replicación previa y corte de 40 min
```

La validación.

```
tráfico en espejo antes de cada corte                     3 días
  diferencias detectadas                                       19
  de ellas, esperadas                                          14
  de ellas, defectos reales                                     5
    → 2 de redondeo, 2 de zona horaria, 1 de orden de resultados
                                                          clase 153

paridad de negocio, tras cada corte
  pedidos por hora, importes y conversión, comparados 7 días
  desviaciones investigadas                                    3
  de ellas, causadas por la migración                          1

plazo de observación por oleada                           1 mes
  → para que corrieran los procesos mensuales
  incidencias detectadas en ese plazo                          6
    → 4 eran procesos periódicos que nadie había inventariado
```

Cuatro de las seis incidencias tardías eran trabajos periódicos, exactamente lo que el apartado cuarto anticipa.

Cómo se midió el programa.

```
cargas                                        41              26
  retiradas                                    —              12
  migradas                                     —              26
  mantenidas con fecha                         —               3
coste mensual del patrimonio                9.400 €        3.100 €
coste por unidad de negocio                    —             −54 %
latencia p99 de las cargas migradas       comparada          −22 %
incidentes causados por la migración           —                2
  → 1 oleada revertida, 1 licencia detectada tarde
cargas en «mantener» con fecha vencida         —                0
centro de datos                             activo          cerrado
                                                            (mes 10)
```

Y la cifra que el equipo consideró más significativa:

```
cargas retiradas                                          12
cargas que se habrían migrado sin inventariar             12
semanas de trabajo evitadas                               ~9
coste mensual evitado                                1.840 €
```

La lección que esta clase traslada a la parte 13: doce de cuarenta y una cargas no había que migrarlas: había que apagarlas, y ninguna aparecía como candidata en el inventario recibido; se descubrieron apagándolas un martes y viendo quién llamaba. Y la única oleada que hubo que revertir se salvó por dos decisiones tomadas antes del corte: un criterio de vuelta atrás escrito y no haber cruzado todavía el punto de no retorno, porque el destino aún no escribía datos propios.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-13-multicloud-hybrid-disaster-recovery/167-las-7r-de-migracion-y-oleadas/lab.py
```

El laboratorio selecciona el motor de práctica migration 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 roadmap-migracion 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 inventario, dependencias, riesgo y oleadas. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye roadmap-migracion 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 migra todo y no se obtiene ningún beneficio	Se eligió una sola forma —rehospedar— para todas las cargas	Elige por carga: retirar, mantener con fecha, rehospedar con plan posterior, cambiar de plataforma, comprar o reescribir.
Aparecen sistemas y dependencias durante el corte	El inventario venía de una lista y el descubrimiento duró poco	Inventaría desde la facturación y el tráfico, observa conexiones durante semanas y apaga a propósito lo que parece no usarse.
Una migración sale mal y no se puede volver atrás	La oleada era demasiado grande o ya se habían escrito datos en el destino	Oleadas pequeñas, criterio de vuelta atrás escrito antes del corte y punto de no retorno declarado.
Durante la convivencia aparecen datos inconsistentes	Origen y destino escriben los mismos datos	Un solo escritor: mueve los datos primero, o deja la escritura en el origen mientras la carga se traslada.
Una carga funciona peor tras migrarla	El rendimiento del destino es distinto y la aplicación era sensible a la latencia	Mide antes, prueba con tráfico en espejo y corrige patrones conversadores antes de mover.
Se declara terminada la migración y semanas después fallan cosas	No corrieron los procesos mensuales y trimestrales dentro del plazo de observación	No cierres una oleada hasta que pase un ciclo completo de negocio.

Seguridad, ética y costo

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

Preguntas de comprobación

1. ¿Cuál de las siete formas tiene el mayor rendimiento y por qué se salta?
2. ¿De dónde debe salir el inventario y qué dos preguntas ahorran más trabajo?
3. ¿Qué tres criterios ordenan las oleadas?
4. ¿Cuáles son las tres formas de resolver quién escribe durante la convivencia?
5. ¿Por qué no se cierra una oleada el mismo día del corte?

Referencias

- AWS (2025). Migration strategies: the 7 Rs — definición y criterios de elección por carga. <https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-guide/migration-strategies.html>
- Google Cloud (2025). Migration to Google Cloud: assess and discover — descubrimiento y dependencias observadas. <https://cloud.google.com/architecture/migration-to-gcp-getting-started>
- Microsoft (2025). Cloud Adoption Framework: migrate — oleadas, ventanas y validación. <https://learn.microsoft.com/azure/cloud-adoption-framework/migrate/>
- Fowler, M. (2004). Strangler fig application — sustitución progresiva con fachada. <https://martinfowler.com/bliki/StranglerFigApplication.html>
- Humble, J. y Farley, D. (2010). Continuous Delivery, cap. 12 — migraciones de datos y convivencia. <https://www.oreilly.com/library/view/continuous-delivery-reliable/9780321670250/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 13 en PDF · Manual integral

Anterior	Índice	Siguiente
← 166 · Backup, RTO, RPO y patrones de disaster recovery	Parte 13 · Programa	168 · Proyecto: continuidad activa-pasiva entre nubes →

Evaluación — 167 Las 7R de migración y oleadas

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 roadmap-migracion 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.

168 — Proyecto: continuidad activa-pasiva entre nubes

Parte: 13 — Multi-cloud, híbrido, migración y recuperación Nivel: experto · Horas estimadas: 8 Laboratorio: capstone · Estado: EXECUTABLECORE

Propósito

Montar la continuidad entre proveedores de verdad —activo-pasivo, que es lo que la parte ha ido justificando como suficiente— y ejecutarla, que es lo único que la convierte en algo real. La clase da el proyecto completo, sus pruebas negativas y las cifras que hay que medir. Y cierra la parte con las tres piezas de siempre: calificar las cinco predicciones de la clase 156, incorporar la ley que esta parte ha hecho aparecer seis veces y que llevaba implícita desde la clase 088, y escribir la predicción que la parte 14 tendrá que corregir.

Resultados de aprendizaje

Al finalizar podrás:

1. Montar una configuración activo-pasivo entre proveedores con lo construido en la parte.
2. Ejecutar una conmutación real y medir sus cinco tramos.
3. Calificar las cinco predicciones de la clase 156 con evidencia.
4. Incorporar la ley 22 al cuestionario, con su recuento histórico.
5. Escribir la predicción de la parte 14 en términos que se puedan desmentir.

Conceptos centrales

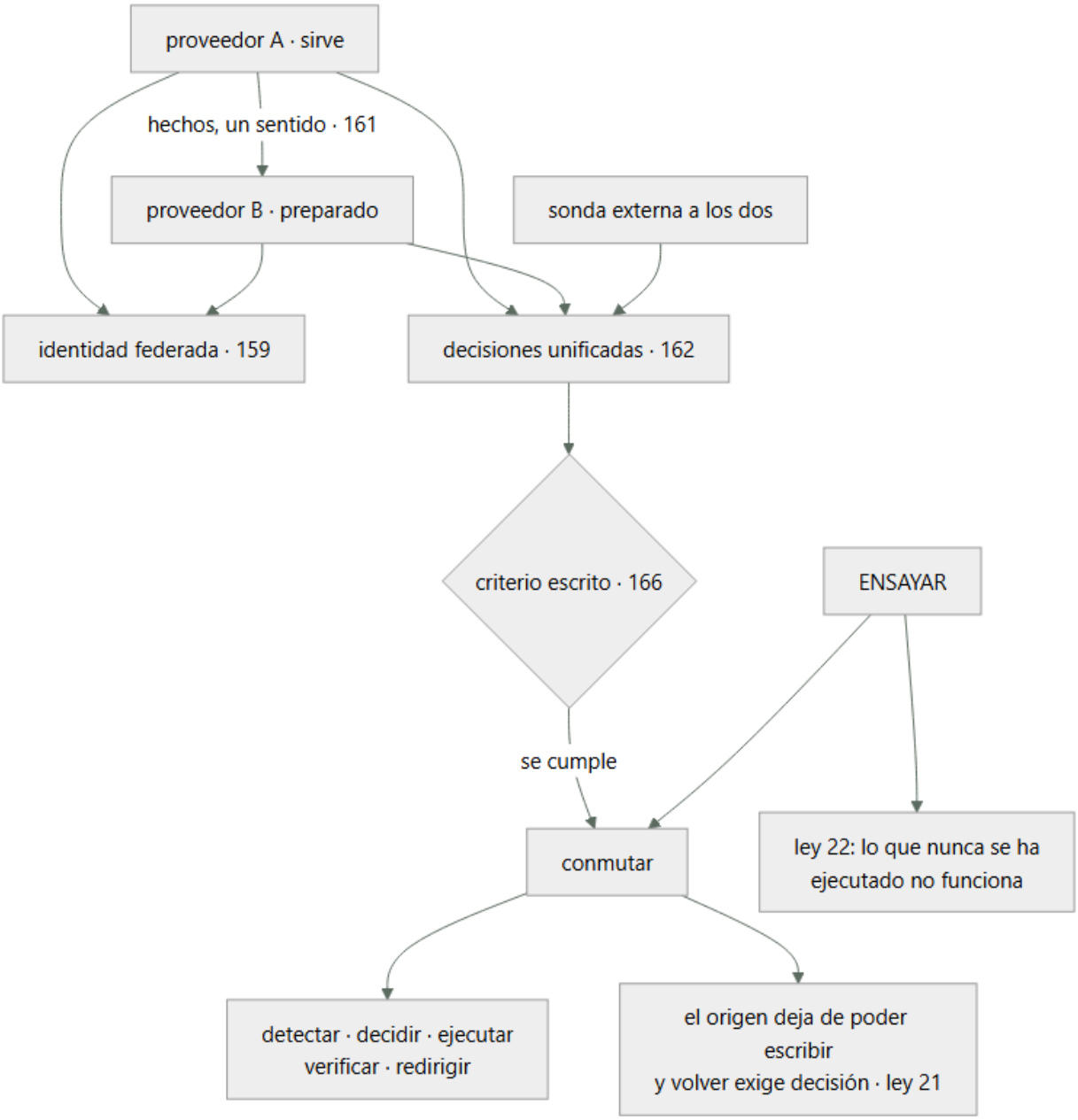
Concepto	Comprensión verificable
activo-pasivo entre proveedores	Un proveedor sirve y el otro está preparado para asumir, con datos replicados y decisión humana.
conmutación ensayada	Ejecución real del plan, con cronómetro y con las consecuencias asumidas. Es lo único que produce una cifra fiable.
ley 22	Un procedimiento que nunca se ha ejecutado no funciona: la primera ejecución siempre encuentra algo, y encontrarlo durante una crisis cuesta mucho más.
calificación de hipótesis	Comparar lo predicho con lo ocurrido publicando también lo que se predijo mal.
coste de estar preparado	Lo que cuesta mantener la opción de conmutar, frente al coste del suceso por su probabilidad.
hipótesis de la parte 14	Predicción escrita ahora sobre lo que ocurrirá cuando el sujeto sea la escala organizativa y el cierre del programa.

Modelo mental

Multi-cloud es una decisión de negocio y riesgo; duplicar cada componente entre proveedores rara vez es la forma más confiable o económica de lograrla.

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. Lo que la parte deja montado

Las once clases anteriores forman un recorrido con una conclusión clara sobre el nivel:

```
motivos interrogados; sobreviven pocos, y ninguno pide activo-activo
                                                        clase 157
portabilidad medida en semanas, no anulada con una capa
                                                        clase 158
identidad federada, sin claves entre nubes               clase 159
conectividad: la mínima, y a menudo ninguna              clase 160
datos: un solo escritor, replicando hechos               clase 161
operación: decisiones unificadas, datos locales          clase 162
infraestructura: un módulo por proveedor, estados separados
                                                        clase 163
flota: idéntico lo que importa, ventana de versiones     clase 164
emplazamientos propios: dueños de sus datos              clase 165
continuidad: objetivos por escenario, medidos            clase 166
migración: retirar primero, oleadas reversibles          clase 167
```

Y el nivel que sale de todo eso, con el vocabulario de la clase 157:

```
nivel 1 para las cargas independientes
nivel 3 —activo-pasivo— para lo que debe sobrevivir a perder un
  proveedor entero
y nivel 5 descartado, con el motivo escrito                ley 21
```

Y las cuatro condiciones sin las cuales el nivel 3 no funciona:

```
1. UN SOLO ESCRITOR en todo momento
   y un mecanismo que impida que el origen vuelva a escribir
   sin decisión explícita                                clase 150

2. CUANTO NO SE REPLICA, PREPARADO DE ANTEMANO
   certificados, secretos, cuotas, permisos, direcciones autorizadas
                                                         clase 166

3. CRITERIO DE DECISIÓN ESCRITO, con quién decide y suplente

4. ENSAYOS PERIÓDICOS
   sin ellos, lo anterior es un documento                ley 22
```

Y el coste de estar preparado, que hay que declarar junto al beneficio:

```
replicación continua de hechos                          ~25-140 €/mes
infraestructura mínima encendida en el destino          ~310 €/mes
cuotas reservadas                                            0 €
ensayos: 4 al año × medio día de tres personas          ~6 días/año
                                                        ■■■■■■■■■■■■
frente a la pérdida estimada de un día de caída          ~90.000 €
```

2. El proyecto

Montar activo-pasivo entre dos proveedores para el flujo de compra, y ejecutarlo. Lo que hay que entregar:

```
1. IDENTIDAD
   federación de personas y de cargas, con sujeto acotado  clase 159
   acceso de emergencia por proveedor, ensayado            clase 134

2. DATOS
   replicación de hechos en un sentido                     clase 161
   copias inmutables en cuenta separada                    clase 166
   y el mecanismo que impide dos escritores                 ley 21

3. INFRAESTRUCTURA
   módulos por proveedor con la misma interfaz             clase 163
   estados separados, cada uno en su proveedor
   destino declarado y creable en minutos

4. RED Y NOMBRES
   tiempos de vida cortos en los registros                 clase 160
   direcciones del destino autorizadas en los terceros

5. OPERACIÓN
   decisiones unificadas y datos locales                   clase 162
   sonda externa a los dos proveedores
   criterio de conmutación escrito, con nombres            clase 166

6. ENSAYO
   conmutación completa, con cronómetro, cada trimestre
   y vuelta atrás, ensayada aparte
```

Y las preguntas cuya respuesta hay que escribir:

```
¿cuánto tarda cada tramo de la conmutación, medido?
¿cuántos datos se pierden, medido?
¿quién decide y quién es su suplente?
¿qué pasa si el origen revive solo?
¿la cuota del destino da para toda la carga?
¿el procedimiento vive fuera del proveedor que puede caerse?
¿cuántas personas pueden ejecutarlo?
```

Las pruebas negativas de la parte 13, que son la entrega más valiosa:

- conmutar de verdad y cronometrar los cinco tramos
- volver atrás y comprobar que no hay dos escritores
- cortar el enlace entre proveedores 15 minutos clase 160
- restaurar una copia y cronometrar clase 166

```
■ simular un borrado y recuperar de copia inmutable
■ desplegar en el segundo proveedor desde cero          clase 158
■ usar el acceso de emergencia de cada proveedor        clase 159
■ comprobar que la sonda externa avisa con el principal caído
■ intentar obtener credenciales de B desde una carga no autorizada
■ dejar sin sincronizar un clúster y ver si salta la alerta  clase 164
■ desplegar una versión que no arranca en un emplazamiento   clase 165
```

Y la advertencia que la ley 22 impone: si alguna no se ha ejecutado nunca, no está resuelta, esté escrita como esté.

3. Calificación de las cinco predicciones

Predicción 1: «menos de la mitad de los motivos de multi-nube sobrevivirán a la pregunta, y el que más sobrevive no será el más citado».

```
veredicto: ACERTADA en las dos mitades

motivos declarados                                      8
sobreviven                                              3   (37,5 %)
el más citado                    «no depender de un proveedor»
  → retirado: se sustituyó por coste de salida medible
los que sobreviven               normativa, capacidad exclusiva
                                 y exigencia de un cliente
```

Predicción 2: «el coste dominante será la salida de datos, con dos cifras porcentuales».

```
veredicto: ACERTADA en la cifra, EQUIVOCADA en el origen

coste de red frente al cómputo                          15 %   ✓

y el desglose:
  tráfico entre zonas del MISMO proveedor y región       31 %
  respuestas a usuarios                                  21 %
  telemetría enviada fuera                               14 %
  …
  replicación entre proveedores                      25 €/mes
```

La partida mayor no cruzaba ninguna frontera entre nubes, y la replicación entre proveedores —lo que la predicción señalaba— resultó ser la más barata de todas, porque mover hechos cuesta el ritmo de cambio y no el tamaño.

Predicción 3: «la ley 21 dominará, y activo-activo con escritura en los dos lados casi nunca compensará».

```
veredicto: ACERTADA

ley 21 en la parte 13                                   5
  157  el nivel 5 se descarta porque el dato tendría dos escritores
  161  el sentido de la replicación se decide por el escritor único
  165  cada emplazamiento es dueño de sus datos
  166  la vuelta atrás y el riesgo de dos escritores
  167  quién escribe durante la convivencia de una migración

y la formulación que quedó:
  si los datos se pueden partir de modo que cada uno tenga un
  escritor, no es activo-activo: son conjuntos independientes
```

Predicción 4: «lo más difícil será la identidad y la red, no mover la carga».

```
veredicto: PARCIALMENTE ACERTADA
```

Identidad y red sí dieron más problemas que la portabilidad:

```
identidad   3 personas de baja con acceso, 180 cargas que podían
            obtener credenciales ajenas, ningún rol común posible
red         4 solapes de rango, un nombre con dos significados,
            una dependencia que se creía blanda
portabilidad  5 fallos al año, detectados por una prueba mensual
```

Y lo que la predicción no vio: el problema mayor de la parte no fue ninguno de los tres. Fue que casi nada se había ejecutado nunca:

```
plan de continuidad         nunca ejecutado → 10 h 51 frente a 4 h
prueba de portabilidad      no existía → 23 fallos acumulados
acceso de emergencia de B   no funcionaba
sonda externa               vivía en el proveedor que se cae
restauración de copia       una vez en 14 meses
```

Predicción 5: «los objetivos de recuperación declarados serán entre 3 y 10 veces menores que los medidos».

```
veredicto: ACERTADA en el fenómeno, EQUIVOCADA en el rango

declarado                                                  4 h
medido                                                10 h 51
factor                                                    ×2,7
```

Por debajo del rango previsto. Y el motivo del error es un acierto de partes anteriores:

```
el tramo de DETECCIÓN, que en una organización sin la parte 10
habría durado horas, duró 90 segundos                      clase 162
→ sin ese trabajo previo, el factor habría estado dentro del rango
```

Recuento de la calificación:

```
acertadas del todo                                                2
acertadas con el fenómeno y falladas en el detalle                3
falladas del todo                                                 0
```

4. La ley 22, el recuento y la hipótesis de la parte 14

```
LEY 22
  Un procedimiento que nunca se ha ejecutado no funciona.
  No es que pueda fallar: es que la primera ejecución SIEMPRE
  encuentra algo, y encontrarlo durante una crisis cuesta un orden
  de magnitud más.
```

Sus seis apariciones en esta parte:

```
clase 157   plan de continuidad de 2023, nunca ejecutado
clase 158   sin prueba de portabilidad: 23 fallos acumulados frente
            a 5 al año con prueba mensual
clase 159   el acceso de emergencia del segundo proveedor no funcionaba
clase 162   la sonda vivía en el proveedor cuya caída debía detectar
clase 165   el camino manual de recuperación: 9 preguntas la primera vez
clase 166   el plan declaraba 4 h y medía 10 h 51
```

Y su historia previa, porque llevaba implícita desde el principio:

```
clase 088   una copia no restaurada no es una copia
clase 102   desplegar una versión rota para comprobar que el canario para
clase 131   el botón de parada de experimentos, que no funcionaba
clase 134   ensayo anual del acceso de emergencia
clase 144   once pruebas negativas; tres fallaron la primera vez
```

Y su diferencia con la ley 13, que es su vecina:

```
ley 13   algo que FUNCIONABA deja de funcionar y no da error
ley 22   algo que NUNCA ha funcionado se supone que funciona
```

Y lo que añade al cuestionario:

```
¿esto se ha ejecutado alguna vez? ¿cuándo? ¿quién?
¿qué encontró la última ejecución?
¿cada cuánto se repite?
¿quién puede ejecutarlo, además de quien lo escribió?
```

Recuento tras la parte 13:

```
ley 13  el bucle que no corre no da error                        27
ley 16  un control que estorba acaba desactivado o rodeado       22
ley 21  el acoplamiento está en quién escribe                     9
ley 15  una señal con demasiados elementos deja de ser señal     20
ley 14  las decisiones de creación son irreversibles             19
ley 20  lo que no tiene dueño no se apaga ni se corrige          11
ley 11  lo que entra en un sistema de solo-añadir se queda        9
ley 22  lo que nunca se ha ejecutado no funciona                  11
        NUEVA, con seis apariciones en esta parte y cinco previas
ley 19  lo que compensa un fallo lo vuelve invisible              7
ley 17  la medida que se vuelve objetivo se alcanza sin mejorar   6
ley 18  lo asíncrono traslada la garantía, no la elimina          5
```

La hipótesis de la parte 14. La parte siguiente sube de escala: zonas de aterrizaje empresariales, gobierno federado, plataforma como producto, modelo operativo, madurez, cargas de inteligencia artificial, soberanía, y el proyecto final. La predicción, escrita para poder desmentirla:

```
1. La escala organizativa no traerá problemas técnicos nuevos:
   traerá los mismos con más gente delante.
   → predigo que la ley dominante será la 16 —el control que estorba
```

```
     acaba rodeado—, porque a escala hay más gente a la que estorbar
   → y que las clases 169 a 174 tratarán, en el fondo, de cómo
     mantener un control cuando quien lo sufre no conoce a quien lo puso

2. En las cargas de inteligencia artificial, el problema dominante
   NO será el modelo ni las tarjetas gráficas: será el DATO.
   → predigo que la clase 175 acabará tratando de propiedad,
     procedencia, residencia y coste de mover datos, y que las leyes
     que aparecerán serán la 14 y la 21, no ninguna nueva

3. En soberanía y computación confidencial, la conclusión honesta
   será que la mayor parte de lo que se vende como soberanía
   resuelve una amenaza que casi nadie tiene.
   → y que lo que sí resuelve un problema real es lo mismo de la
     clase 141: saber dónde está cada dato y poder demostrarlo

4. El proyecto final encontrará que lo que falta no es tecnología:
   → predigo que la mayoría de sus hallazgos serán de las leyes 20
     y 22 —cosas sin dueño y cosas nunca ejecutadas—, y no fallos
     de diseño

5. Y la predicción que puede salir del revés: al recorrer el programa
   entero, las leyes que más habrán aparecido no serán las más útiles.
   → predigo que la 13 y la 15 dominarán el recuento y que las que
     más decisiones habrán cambiado serán la 14 y la 21
```

Y lo que se anota para calificar sin trampa:

```
lo que ya sabemos    que la 13 y la 15 dominan el recuento
lo que creemos       que el valor de una ley no está en su frecuencia
lo que no sabemos    si hay alguna ley que sobra, o que es un caso
                     particular de otra
```

Ejemplo trabajado

CloudShop monta activo-pasivo entre sus dos proveedores para el flujo de compra y lo ejecuta cuatro veces en un año. Lo que sigue es el resultado de cada ensayo y el recuento de la parte.

El montaje.

```
proveedor A          sirve el 100 % del tráfico
proveedor B          preparado, con lo mínimo encendido
replicación          hechos, un sentido, ~2,1 s de retardo   clase 161
coste de estar preparado                                  ~480 €/mes
  replicación                                                140 €
  infraestructura mínima                                     310 €
  copias inmutables en cuenta separada                        30 €
```

Y lo preparado de antemano, de la lista de la clase 166:

```
cuotas del destino                  150 instancias, solicitadas
certificados                        emitidos y replicados
secretos                            presentes, con identidad propia
permisos y roles                    equivalentes, comprobados  clase 159
direcciones autorizadas en terceros las de B, añadidas
registro de imágenes                réplica en B
tiempos de vida de nombres          60 s
procedimiento                       repositorio replicado + impreso
trabajos periódicos                 declarados, desactivados en B
```

Ensayo 1.

```
detectar                                  80 s
decidir                                    4 min
ejecutar                                  14 min
verificar                                  3 min
redirigir                                  9 min
                                          ■■■■■■
total                                     31 min
pérdida de datos                           22 s

hallazgos                                      3
  los trabajos periódicos no se activaron en B: 2 no corrieron
  una alerta seguía apuntando a recursos de A
```

```
  y el panel de objetivos mezclaba datos de los dos
```

Ensayo 2, tres meses después.

```
total                                      24 min
pérdida                                     18 s
hallazgos                                      2
  una carga nueva desplegada en el trimestre no estaba replicada
    → nadie había añadido su base a la replicación
  un certificado emitido en el trimestre no se había copiado a B
```

Y la corrección fue de proceso, no técnica:

```
toda carga nueva pasa por una comprobación de continuidad
  ¿sus datos se replican?
  ¿sus secretos y certificados están en el destino?
  ¿está en el procedimiento?
→ una casilla en la plantilla de servicio nuevo         clase 106
cargas sin cobertura detectadas después                        0
```

Ensayo 3: la vuelta atrás.

```
se conmutó y se permaneció en B durante 6 horas, sirviendo tráfico real
y después se volvió

conmutar                                   22 min
permanencia en B                            6 h
  latencia p99                             +18 ms
  incidencias                                   1  (una integración
                                                   con lista de direcciones
                                                   que no se había
                                                   actualizado)
volver                                     48 min
  replicación en sentido inverso            31 min
  comprobación de escritor único            ✓
  pedidos escritos en el sitio equivocado         0
```

Seis horas sirviendo tráfico real desde el segundo proveedor, que es la única forma de saber que funciona.

Ensayo 4: con el proveedor principal realmente degradado.

No fue un ensayo: fue un incidente real del proveedor.

```
11:02  degradación del proveedor A en la región principal
11:03  la sonda externa avisa                                    60 s
11:07  se cumple el criterio escrito; decide quien está de guardia
11:21  conmutado y verificado
11:30  tráfico redirigido

total                                      28 min
pérdida                                     31 s
intervención de quien construyó el sistema          ninguna
personas capaces de ejecutarlo                    6 de 9
```

Y la comparación que cierra la parte:

```
primer ensayo, hace un año                 10 h 51
incidente real                             28 min
factor                                     ×23 más rápido
y nada de esa mejora vino de cambiar el patrón: sigue siendo
  activo-pasivo con lo mínimo encendido
```

Las once pruebas negativas de la parte.

```
 1. conmutar y cronometrar                    ✓  4 ejecuciones
 2. volver atrás sin dos escritores           ✓
 3. cortar el enlace entre proveedores        ✓  (clase 160)
 4. restaurar una copia y cronometrar         ✓  38 min
 5. simular un borrado y recuperar            ✓  pérdida 5 min
 6. desplegar en B desde cero                 ✓  mensual
 7. acceso de emergencia de cada proveedor    ✓  falló el primero
 8. sonda externa con el principal caído      ✓  falló el primero
 9. credenciales desde una carga no autorizada ✓ falló el primero
10. clúster sin sincronizar                   ✓  falló el primero
11. versión que no arranca en un emplazamiento ✓ falló el primero

pruebas que fallaron la primera vez                     5 de 11
```

Cinco de once fallaron la primera vez que se ejecutaron, y las cinco estaban documentadas como resueltas. Es la ley 22, medida.

El recuento de la parte 13.

```
proveedores                                3              2
motivos escritos e interrogados            0              8
motivos vivos                              —              3
nivel de multi-nube                  sin decidir          1 y 3
capa de abstracción propia            propuesta      no existe
coste de salida documentado               no             sí
usuarios locales en las nubes             15             0
claves de larga duración entre nubes       3             0
conectividad privada entre proveedores  proyecto          0
solapes de rango                           4             0
coste mensual de red                   2.850 €         520 €
coste de observar el segundo proveedor   728 €         219 €
estados que abarcan dos proveedores        3             0
clústeres                                  9             6
tiendas con operación sin conexión   340 de 340   340 de 340
plazo de recuperación declarado          4 h          45 min
plazo de recuperación medido          10 h 51         28 min
copias inmutables en cuenta separada       no            sí
ensayos de conmutación ejecutados          0             4
personas capaces de ejecutar la conmutación  2           6
cargas migradas y retiradas                —      26 y 12
```

Lo que la parte 13 no resolvió, dicho con claridad.

```
el coste de salida del servicio de análisis sigue siendo alto
  y se ha aceptado por escrito, con revisión anual
los metadatos siguen procesándose fuera de la región exigida
  documentado, no resuelto                              clase 141
y la vuelta atrás sigue tardando más que la conmutación,
  porque el destino acumula datos nuevos
```

La conclusión que cierra la parte 13: el plan de continuidad existía desde hacía dos años y declaraba cuatro horas; al ejecutarlo por primera vez tardó casi once, y de ese tiempo el 41 % no fue ejecutar nada. Un año y cuatro ensayos después, un incidente real del proveedor se resolvió en veintiocho minutos sin que interviniera nadie de quienes lo construyeron. El patrón no cambió: seguía siendo activo-pasivo con lo mínimo encendido. Lo que cambió fue haberlo ejecutado.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-13-multicloud-hybrid-disaster-recovery/168-proyecto-continuidad-activa-pasiva-entre-nubes/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 plataforma-multicloud 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 plataforma-multicloud 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
Existe un plan de continuidad y nadie sabe cuánto tarda de verdad	Ley 22: nunca se ha ejecutado	Ejecútalo con cronómetro, mide los cinco tramos y repítelo cada trimestre.
Una carga nueva no está cubierta por el plan	La continuidad no forma parte del alta de un servicio	Añade una comprobación de continuidad a la plantilla de servicio nuevo: replicación, secretos, certificados y procedimiento.
Tras conmutar, el origen revive y acepta escrituras	No hay mecanismo que se lo impida	Marca el origen como no autorizado a escribir al conmutar y exige decisión explícita para volver.
Se monta activo-activo con escritura en los dos lados y aparecen conflictos	Un dato con dos escritores	Parte los datos por cliente o por región para que cada uno tenga un escritor; si no se puede, activo-pasivo.
Los ensayos se hacen sin tráfico real y no demuestran nada	Se ejecuta el procedimiento sin permanecer en el destino	Permanece sirviendo tráfico real varias horas y vuelve después; ahí aparecen las integraciones que faltaban.
Se dan por buenas las predicciones de la parte anterior sin revisarlas	Calificar solo lo que salió bien convierte el aprendizaje en opinión	Publica el veredicto de cada una con su evidencia, incluidas las tres que acertaron el fenómeno y fallaron el detalle.

■ 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 condiciones hacen falta para que activo-pasivo funcione?
2. ¿En qué acertó y en qué falló la predicción sobre el coste de salida de datos?
3. ¿Por qué el factor entre el plazo declarado y el medido fue menor de lo predicho?
4. ¿Qué dice la ley 22 y en qué se diferencia de la ley 13?
5. ¿Qué predice la hipótesis de la parte 14 sobre las cargas de inteligencia artificial y sobre la soberanía?

Referencias

- AWS (2025). Disaster recovery across providers: considerations — límites y coste de la continuidad entre nubes. <https://docs.aws.amazon.com/whitepapers/latest/disaster-recovery-workloads-on-aws/disaster-recovery-workloads-on-aws.html>
- Google Cloud (2025). Architecting disaster recovery for cloud infrastructure outages — objetivos, patrones y pruebas. <https://cloud.google.com/architecture/disaster-recovery>
- Azure (2025). Reliability testing and failover drills — ensayos periódicos y su registro. <https://learn.microsoft.com/azure/reliability/>
- Google SRE (2025). Disaster role playing and readiness testing — ejecutar el plan como parte de la operación. <https://sre.google/workbook/>
- Uptime Institute (2025). Outage analysis and recovery practice — frecuencia real de fallos por región y por proveedor. <https://uptimeinstitute.com/resources>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 13 en PDF · Manual integral

Anterior	Índice	Siguiente
← 167 · Las 7R de migración y oleadas	Parte 13 · Programa	169 · Landing zones empresariales y vending de cuentas →

Evaluación — 168 Proyecto: continuidad activa-pasiva entre nubes

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 plataforma-multicloud 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.