Multi-Cloud Engineering Program

Parte 16 — Redes cloud avanzadas, conectividad híbrida y edge

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	16 de 23
Clases	193–204
Evaluaciones	12
Horas estimadas	52

Índice

Parte 16 — Redes cloud avanzadas, conectividad híbrida y edge

← Parte 15 · Índice completo · Parte 17 →

Descargar: Esta parte en PDF · Manual integral

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

Diseñar conectividad segura y observable entre regiones, proveedores, centros de datos y edge.

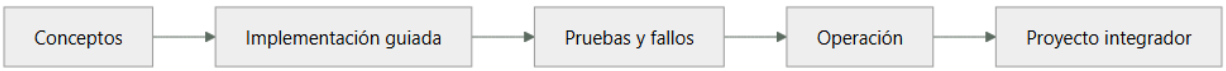
Resultados de la parte

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

La red cloud es un sistema distribuido de rutas, identidades y políticas; cada salto añade latencia, costo y una frontera de fallo.

Secuencia



Clases

ID	Clase	Laboratorio	Horas
193	CIDR, subnetting y planificación IP a escala	network	4
194	Routing, BGP, tránsito y propagación de rutas	network	4
195	DNS autoritativo, recursivo, split-horizon y DNSSEC	network	4
196	Balanceo L4/L7, proxies, TLS y gestión de certificados	network	4
197	CDN, caché, origin shielding y edge compute	performance	4
198	VPN, Direct Connect, ExpressRoute e Interconnect	network	4
199	Transit Gateway, Virtual WAN y Network Connectivity Center	network	4
200	Private endpoints, service networking y egress control	security	4
201	Service mesh, mTLS y gestión de tráfico este-oeste	network	4
202	eBPF, flow logs, packet capture y diagnóstico	observability	4
203	SD-WAN, 5G, IoT y operación desconectada	architecture	4
204	Proyecto: red multi-región y multi-cloud	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

- Computer Networking: A Top-Down Approach — Kurose y Ross.
- Cloud Native Data Center Networking — Dinesh Dutt.
- AWS Advanced Networking Study Guide — S. Mahalingam.

193 — CIDR, subnetting y planificación IP a escala

Parte: 16 — Redes cloud avanzadas, conectividad híbrida y edge Nivel: avanzado · Horas estimadas: 4 Laboratorio: network · Estado: EXECUTABLECORE

Propósito

Diseñar un plan de direccionamiento que aguante diez años, porque es la decisión que más veces se toma en una tarde y menos veces se puede deshacer. La clase da la aritmética de CIDR sin rodeos, el método de asignación jerárquica que evita el solapamiento, la lista de consumidores de direcciones que siempre se olvidan, y el procedimiento de renumeración para cuando ya es tarde.

Resultados de aprendizaje

Al finalizar podrás:

1. Calcular rangos, máscaras y capacidad de un bloque CIDR sin herramientas.
2. Asignar direcciones jerárquicamente por región, entorno y función.
3. Dimensionar contando los consumidores que no se ven.
4. Evitar el solapamiento que impide interconectar más adelante.
5. Renumerar una red en producción cuando el plan se agotó.

Conceptos centrales

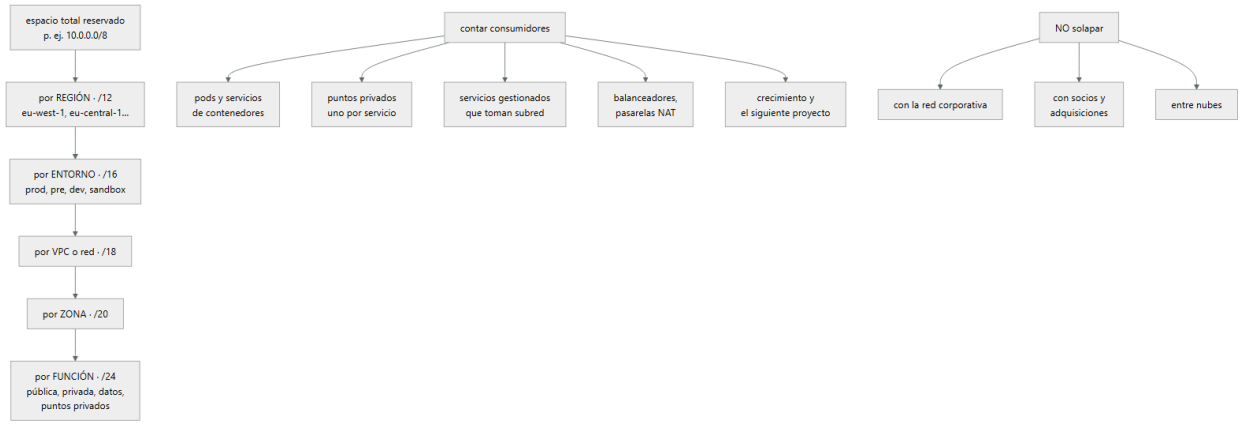
Concepto	Comprensión verificable
CIDR	Notación de bloque de direcciones: prefijo y longitud. /24 son 256 direcciones, /16 son 65.536.
asignación jerárquica	Repartir un bloque grande en trozos por región, entorno y función, de forma que cada nivel resuma en una sola ruta.
solapamiento	Dos redes que usan el mismo rango. Impide conectarlas sin traducción y es el error más caro.
resumen de rutas	Anunciar un bloque grande en vez de muchos pequeños. Depende de que la asignación sea contigua.
consumidor oculto	Lo que gasta direcciones sin que nadie lo cuente: servicios gestionados, contenedores, puntos privados.
renumeración	Cambiar el direccionamiento de una red viva. Cara, larga y evitable con un plan inicial correcto.

Modelo mental

La red cloud es un sistema distribuido de rutas, identidades y políticas; cada salto añade latencia, costo y una frontera de fallo.

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 aritmética, sin herramientas

Un plan de direccionamiento se discute en una pizarra, y para eso hay que saber contar sin calculadora.

```
la longitud del prefijo dice cuántos bits FIJA
32 - longitud = bits libres = direcciones

/8    16.777.216      /20     4.096
/12    1.048.576      /21     2.048
/16       65.536      /22     1.024
/17       32.768      /23       512
/18       16.384      /24       256
/19        8.192      /26        64
                      /28        16
```

Y dos reglas que evitan el 90 % de los errores de cálculo:

```
cada bit menos de prefijo DUPLICA el tamaño
  /24 → /23 → /22 → /21   256, 512, 1.024, 2.048

un bloque siempre empieza en un múltiplo de su tamaño
  10.0.4.0/22 es válido      (4 es múltiplo de 4)
  10.0.5.0/22 NO es válido
```

Y lo que se pierde por bloque, que en subredes pequeñas es mucho:

```
direcciones reservadas por subred
  red y difusión                                  2
  las nubes reservan además entre 3 y 5

  /24  →  ~251 usables
  /28  →  ~11 usables    ← el 31 % perdido
→ por eso no conviene bajar de /26 en subredes de trabajo
```

Y los espacios privados disponibles, que es lo primero que se reparte:

```
10.0.0.0/8          16,7 M    el grande; el que se usa
172.16.0.0/12        1,0 M    ojo: muchos productos lo usan
                              por defecto
192.168.0.0/16      65 K      doméstico; no usar en empresa
100.64.0.0/10        4,2 M    espacio compartido; útil para
                              lo que nunca se enruta
```

Y una advertencia sobre el segundo:

```
172.17.0.0/16 es el rango por defecto de un motor de
contenedores muy extendido
→ usarlo en la red corporativa produce fallos intermitentes
  y difíciles de diagnosticar
```

2. Asignación jerárquica

El plan que funciona reparte de arriba abajo, y su virtud no es la elegancia: es que cada nivel se puede resumir en una sola ruta.

```
NIVEL 0   el espacio de la organización
  10.0.0.0/8, reservado entero aunque se use el 2 %
  → y ANOTADO en un registro central                clase 169

NIVEL 1   por región         /12   (1 M de direcciones)
  10.0.0.0/12    eu-west-1
  10.16.0.0/12   eu-central-1
  10.32.0.0/12   us-east-1
  10.48.0.0/12   reservado
  → una ruta por región en la tabla central

NIVEL 2   por entorno        /16
  10.0.0.0/16    producción
  10.1.0.0/16    preproducción
  10.2.0.0/16    desarrollo
  10.3.0.0/16    sandbox
  10.4-15        reservado para crecer

NIVEL 3   por red o cuenta   /18 o /20

NIVEL 4   por zona           /20 o /22

NIVEL 5   por función        /24
  .0/24   pública (balanceadores, NAT)
  .1/24   privada de aplicación
  .2/24   datos
  .3/24   puntos privados      ← el que siempre falta
```

Y las tres reglas que hacen que el plan sobreviva:

```
1. RESERVA MÁS DE LO QUE NECESITAS
   reservar no cuesta nada; renumerar cuesta meses
   → si crees que necesitas un /20, reserva un /16

2. DEJA HUECOS ENTRE BLOQUES
   asignar consecutivo sin huecos impide ampliar sin partir
   → el bloque siguiente al tuyo debe estar libre

3. QUE CADA NIVEL SEA CONTIGUO
   sin contigüidad no hay resumen de rutas
   → y las tablas de encaminamiento tienen límite duro
```

Y el límite que sorprende a mitad de proyecto:

```
las tablas de rutas de las nubes tienen máximos
  típicamente entre 50 y 1.000 rutas
→ sin resumen, se llega antes de lo que parece
→ y ampliar el límite no siempre es posible
```

Y una decisión que conviene tomar al principio:

```
¿IPv6 desde el día uno?
  a favor   no hay escasez; simplifica el plan; obligatorio
            en algunos servicios nuevos
  en contra doble pila duplica reglas y diagnósticos; no
            todos los productos lo soportan
  práctica  reservar el plan IPv6 aunque no se despliegue,
            y activarlo donde no cueste
```

3. Contar lo que no se ve

El agotamiento de direcciones casi nunca lo causan las máquinas: lo causan los consumidores que nadie contó.

```
LO QUE SE CUENTA
  máquinas virtuales
  bases de datos

LO QUE NO SE CUENTA Y GASTA MÁS
  PODS DE CONTENEDORES
    en algunos modelos de red, cada pod toma una dirección
    de la subred → 110 pods por nodo × 40 nodos = 4.400
    → una subred /24 se agota con 2 nodos

  PUNTOS PRIVADOS
    uno por servicio y por subred                clase 200
```

```
    → 30 servicios × 3 zonas = 90 direcciones

  SERVICIOS GESTIONADOS QUE PIDEN SUBRED PROPIA
    bases gestionadas, motores de análisis, pasarelas
    → algunos exigen un /27 o /26 vacío y exclusivo

  BALANCEADORES Y PASARELAS NAT
    varias direcciones cada uno, por zona

  DESPLIEGUES ESCALONADOS
    durante un despliegue conviven dos versiones
    → hasta el doble de direcciones durante horas  clase 102

  ESCALADO AUTOMÁTICO
    hay que dimensionar por el MÁXIMO, no por el habitual
```

Y la regla de dimensionado que resulta de todo eso:

```
calcula el máximo previsible y MULTIPLICA POR 2
→ y si el resultado es un /24, usa un /23
```

El solapamiento, que es el error más caro y el más frecuente:

```
DÓNDE OCURRE
  dos equipos eligen 10.0.0.0/16 por defecto
  la nueva nube usa el mismo rango que la corporativa
  una empresa adquirida usa el mismo espacio
  un socio pide conexión y coincide
  el proveedor gestionado usa 172.17 por dentro

QUÉ IMPIDE
  conectar las dos redes sin traducción
  y la traducción rompe el diagnóstico, los registros y
  cualquier control basado en origen               clase 135
```

Y la prevención, que es barata:

```
un REGISTRO CENTRAL de bloques asignados
  quién, para qué, desde cuándo, y estado
  y ninguna red se crea sin pedir el bloque ahí
→ automatizado en la plantilla de red nueva      clase 171
→ si pedir el bloque tarda tres días, alguien lo elegirá
  a mano                                            ley 16
```

4. Renumerar, cuando ya es tarde

Antes o después aparece una red que hay que renumerar. Es caro, pero no imposible, y hacerlo mal lo convierte en un corte.

```
CUÁNDO ES INEVITABLE
  el bloque se agotó y no hay contiguo libre
  hay solapamiento con algo que hay que conectar
  una adquisición trae la misma red
```

El procedimiento que evita el corte, que es siempre el mismo patrón de expandir y contraer:

```
1  AÑADIR el rango nuevo a la red existente
   la mayoría de las nubes permiten varios bloques por red

2  CREAR subredes nuevas en el rango nuevo

3  MOVER cargas, una a una, con despliegue escalonado
   → lo que se mueve primero: lo que no tiene dependencias
     entrantes por dirección

4  ACTUALIZAR lo que referencia direcciones
   ← aquí está el trabajo real, y es donde se descubre
     lo que nadie sabía

5  VACIAR y retirar las subredes viejas

6  QUITAR el bloque antiguo de la red
```

Y el paso 4 merece su lista, porque es lo que rompe:

```
reglas de cortafuegos con direcciones fijas
```

```
listas de permitidos de terceros                    ← las peores
configuraciones con la dirección escrita a mano
rutas estáticas y túneles
registros de nombres con direcciones                 clase 195
certificados emitidos para una dirección
controles de acceso a bases por dirección de origen
cuadros de mando y alertas que filtran por rango
```

Y una realidad sobre las listas de terceros:

```
cambiar una dirección que un socio tiene en su lista de
permitidos requiere SU calendario, no el tuyo
→ semanas o meses
→ por eso el tráfico saliente hacia terceros conviene que
  salga siempre por direcciones fijas y pocas, reservadas
  desde el principio
```

Y la lista de comprobación de la clase:

- hay un espacio reservado para toda la organización
- hay un registro central de bloques, y es obligatorio
- la asignación es jerárquica: región, entorno, red, zona, función
- cada nivel es contiguo y se puede resumir
- hay huecos reservados junto a cada bloque asignado
- el dimensionado cuenta pods, puntos privados y servicios gestionados
- el dimensionado usa el máximo, multiplicado por dos
- ninguna subred de trabajo baja de /26
- no se usa 172.17.0.0/16
- se ha comprobado el solapamiento con la red corporativa, socios y otras nubes
- el tráfico saliente hacia terceros usa direcciones fijas y pocas
- el plan IPv6 está reservado aunque no se despliegue
- se conocen los límites de rutas por tabla

Y el cierre que enlaza con la clase siguiente: con las direcciones repartidas, queda decidir cómo se anuncian y por dónde va cada paquete. Encaminamiento, BGP y propagación de rutas es la materia de la clase 194.

Ejemplo trabajado

CloudShop tiene 214 cuentas, tres nubes y una red corporativa de quince años. Lo que sigue es el estado del direccionamiento que se encontró, el plan nuevo, y la renumeración de la red que impedía conectar con la empresa adquirida.

Lo que había, levantado en dos semanas:

```
redes encontradas                                    87
  con bloque registrado en algún sitio               19
  con bloque documentado en una hoja de cálculo      31
  sin registro alguno                                37

solapamientos detectados
  10.0.0.0/16 usado en 6 redes distintas
  192.168.1.0/24 en 11 redes de desarrollo
  172.17.0.0/16 en la red corporativa Y en el motor
    de contenedores → causaba fallos intermitentes
    desde 2022, con 3 incidentes sin causa conocida

la adquisición
  la empresa comprada usa 10.0.0.0/16 para su producción
  → imposible conectar sin traducción
```

Y el diagnóstico de por qué había pasado:

```
no existía registro central obligatorio
y pedir un bloque al equipo de red tardaba 6 días
→ los equipos elegían el rango por defecto de la plantilla
→ es exactamente el mecanismo de la clase 169         ley 16
```

El plan nuevo.

```
espacio reservado          10.0.0.0/8   entero
```

```
NIVEL 1 · región                                        /12
  10.0.0.0/12     eu-west-1        (AWS, principal)
  10.16.0.0/12    eu-central-1     (AWS, secundaria)
  10.32.0.0/12    westeurope       (Azure)
  10.48.0.0/12    europe-west1     (GCP)
  10.64.0.0/12    corporativa
  10.80.0.0/12    adquisiciones    ← reservado a propósito
  10.96.0.0/11    libre

NIVEL 2 · entorno, dentro de cada región                /16
  x.0.0.0/16      producción
  x.1.0.0/16      preproducción
  x.2.0.0/16      desarrollo
  x.3.0.0/16      sandbox
  x.4-15          libre

NIVEL 3 · red por dominio de producto                   /18
  10.0.0.0/18     pedidos
  10.0.64.0/18    catálogo
  10.0.128.0/18   datos y análisis
  10.0.192.0/18   libre

NIVEL 4 · zona                                          /20
NIVEL 5 · función                                       /24
  x.x.0.0/24      pública
  x.x.1.0/23      privada de aplicación   ← /23, ver abajo
  x.x.4.0/24      datos
  x.x.5.0/24      puntos privados
  x.x.6-15        libre
```

El dimensionado, contando lo que no se ve. El cálculo que cambió el plan:

```
subred privada de aplicación, zona de producción de pedidos

  cálculo ingenuo
    40 máquinas + margen           →  /26  (64)

  cálculo real
    nodos de contenedores                        40
    pods por nodo                               110
    direcciones por pods                      4.400
    durante despliegue escalonado (×2)        8.800
    balanceadores internos                       12
    puntos privados (30 servicios × 3 zonas)     90
    bases gestionadas con subred propia       3 × 32
    margen de escalado al máximo              +30 %
    ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
    total por zona                          ~11.900

  → /18 por zona, no /26
  → el plan inicial se habría agotado en 4 meses
```

Y la corrección que se hizo en el plan:

```
se cambió el modelo de red de contenedores para que los
pods usen un rango separado y no enrutado
  100.64.0.0/10 para pods, no enrutado fuera del clúster
  → la subred de aplicación vuelve a caber en un /22
  → y las direcciones enrutables se reservan para lo que
    de verdad necesita ser alcanzable
```

El registro central, montado antes que nada:

```
un bloque no se puede usar sin estar registrado
  quién lo pide, para qué, desde cuándo, estado

y la parte que lo hace funcionar
  la plantilla de red nueva PIDE el bloque automáticamente
  → tiempo de espera: 40 segundos, no 6 días
  → función de aptitud: ninguna red con bloque no
    registrado                                  clase 190

resultado a los 3 meses
  redes creadas                                   23
  con bloque del registro                         23
```

```
  elegidas a mano                                    0
```

La renumeración de la red de producción de la adquisición.

```
situación   10.0.0.0/16, solapando con CloudShop
destino     10.80.0.0/18, dentro del espacio reservado
cargas      41 servicios, 2 bases, 6 túneles a socios

semana 1    añadir 10.80.0.0/18 a la red existente
semanas 2-3 crear subredes nuevas y mover 12 servicios sin
            dependencias entrantes por dirección
semanas 4-8 mover los 29 restantes, de 4 en 4
semanas 3-14 ACTUALIZAR REFERENCIAS   ← el trabajo real
```

Y lo que apareció en el paso de referencias:

```
reglas de cortafuegos con direcciones fijas          214
rutas estáticas                                       31
configuraciones con dirección escrita a mano          88
  de ellas, en ficheros que nadie sabía que existían  19
listas de permitidos de terceros                       9
  plazo del más lento                          11 semanas
certificado emitido para una dirección                 1
controles de base de datos por origen                 17
alertas que filtran por rango                         23

hallazgos no buscados
  4 servicios que se creían retirados seguían recibiendo
    tráfico
  1 túnel a un socio que dejó de operar en 2021       ley 20
  2 reglas de cortafuegos que permitían todo el tráfico
    desde un rango de un proveedor que ya no se usaba
```

Y el resultado:

```
duración total                            14 semanas
cortes de servicio                             0
  → gracias a convivir los dos rangos durante la migración
el plazo lo marcó                      una lista de permitidos
                                       de un socio: 11 semanas
coste estimado                      1,8 personas × 14 semanas
```

La comparación que cierra el caso:

```
reservar el espacio y registrar los bloques al principio
  coste            2 días de trabajo
renumerar después
  coste            25 personas-semana, y el calendario de
                   un tercero
→ y esta es la razón por la que el plan de direccionamiento
  se decide con cuidado y pronto                    ley 14
```

La lección que esta clase deja: el plan inicial habría durado cuatro meses porque nadie contó los pods, y la corrección no fue pedir más direcciones sino sacar los pods del espacio enrutable. Y de las catorce semanas de renumeración, el plazo no lo marcó ninguna decisión técnica: lo marcó la lista de permitidos de un socio, que tenía su propio calendario.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-16-advanced-cloud-networking-edge/193-cidr-subnetting-y-planificacion-ip-a-escala/
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 ipam-plan en evidence/ usando la plantilla indicada.

5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

El artefacto principal es una 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 ipam-plan para el caso CloudShop. Incluye una alternativa descartada, un supuesto que pueda falsarse, una prueba de fallo y una decisión de rollback.

■ Criterio de aceptación

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

■■ Errores frecuentes

Síntoma	Causa probable	Corrección
Dos redes que hay que conectar usan el mismo rango	No hubo registro central obligatorio y cada equipo eligió el valor por defecto	Reserva un espacio para toda la organización, exige registro para crear cualquier red y automatiza la petición para que tarde segundos.
Una subred se agota a los pocos meses	Se dimensionó contando máquinas y no pods, puntos privados ni servicios gestionados	Cuenta todos los consumidores, dimensiona por el máximo y multiplica por dos; saca los pods a un rango no enrutado si el modelo lo permite.
No se puede ampliar un bloque sin partirlo en dos	Los bloques se asignaron consecutivos, sin huecos	Deja libre el bloque contiguo a cada asignación y reserva por nivel más de lo que se necesita.
La tabla de rutas alcanza su límite	La asignación no es contigua y no se puede resumir	Asigna jerárquicamente y de forma contigua para anunciar un bloque por nivel.
Fallos intermitentes de red imposibles de diagnosticar	Un rango usado internamente por un producto solapa con la red corporativa	Evita 172.17.0.0/16 y comprueba qué rangos usan por dentro los productos gestionados antes de asignar.
Una renumeración se alarga meses más de lo previsto	Hay direcciones escritas en listas de permitidos de terceros	Haz que todo el tráfico saliente hacia terceros use direcciones fijas y pocas, reservadas desde el principio.

■ 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. ¿Cuántas direcciones tiene un /22 y por qué 10.0.5.0/22 no es válido?
2. ¿Qué tres reglas hacen que un plan de direccionamiento sobreviva?

3. ¿Qué consumidores de direcciones se olvidan casi siempre?
4. ¿Por qué el solapamiento es el error más caro?
5. ¿Cuál es el paso de una renumeración donde está el trabajo real?

Referencias

• RFC 4632 — Classless Inter-Domain Routing (CIDR). <https://www.rfc-editor.org/rfc/rfc4632>

• RFC 1918 — Address allocation for private internets. <https://www.rfc-editor.org/rfc/rfc1918>

• RFC 6598 — IANA-reserved IPv4 prefix for shared address space (100.64.0.0/10). <https://www.rfc-editor.org/rfc/rfc6598>

• AWS (2025). VPC sizing and IP address planning. <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-cidr-blocks.html>

• Microsoft (2025). Plan for IP addressing in Azure landing zones. <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/plan-for-ip-addressing>

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

Descargar: Parte 16 en PDF · Manual integral

Anterior	Índice	Siguiente
← 192 · Proyecto: arquitectura completa de CloudShop	Parte 16 · Programa	194 · Routing, BGP, tránsito y propagación de rutas →

Evaluación — 193 CIDR, subnetting y planificación IP a escala

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

Criterio de aceptación

• [ ] Resultado reproducible por una segunda persona.

• [ ] Evidencia vinculada a cada afirmación técnica.

• [ ] Prueba negativa o de fallo incluida.

• [ ] Costos y permisos expresados con unidades y alcance.

• [ ] Limitaciones declaradas sin presentar la simulación como producción.

Escala

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

194 — Routing, BGP, tránsito y propagación de rutas

Parte: 16 — Redes cloud avanzadas, conectividad híbrida y edge Nivel: avanzado · Horas estimadas: 4 Laboratorio: network · Estado: EXECUTABLECORE

Propósito

Entender por dónde va cada paquete y por qué, que es la pregunta que nadie sabe contestar durante los incidentes de red. La clase explica cómo se elige una ruta —longitud de prefijo primero, y solo después las preferencias—, qué es BGP y qué garantiza realmente, cómo se propagan las rutas entre la nube y la red corporativa, y por qué el mecanismo que anuncia una ruta es también el que la retira mal.

Resultados de aprendizaje

Al finalizar podrás:

1. Predecir qué ruta elegirá un paquete dadas varias opciones.
2. Explicar qué decide BGP y qué no.
3. Controlar la propagación de rutas entre nube y red corporativa.
4. Diagnosticar los fallos de encaminamiento más comunes.
5. Diseñar la salida y la vuelta del tráfico de forma simétrica.

Conceptos centrales

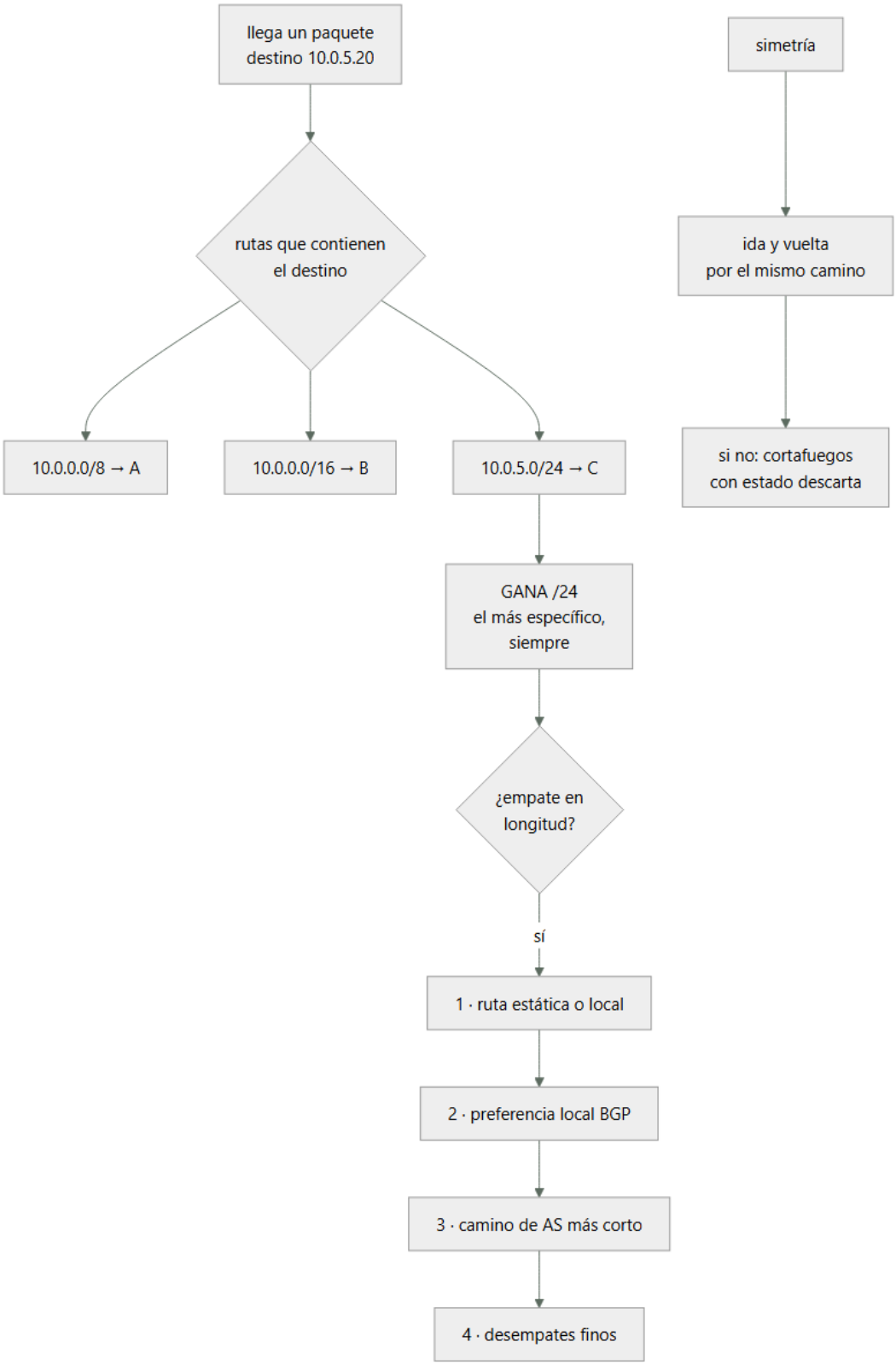
Concepto	Comprensión verificable
prefijo más específico	La ruta con máscara más larga gana siempre, antes que cualquier preferencia.
BGP	Protocolo con el que dos redes se anuncian qué prefijos alcanzan. Es de política, no de rendimiento.
sistema autónomo (AS)	Conjunto de redes bajo una misma política de encaminamiento, identificado por un número.
propagación	Difusión de una ruta aprendida hacia otras tablas. Lo que se propaga sin control acaba donde no debe.
asimetría	Que la ida y la vuelta usen caminos distintos. Rompe cortafuegos con estado y balanceadores.
agujero negro	Ruta que atrae tráfico hacia un destino que no lo entrega. Peor que no tener ruta.

Modelo mental

La red cloud es un sistema distribuido de rutas, identidades y políticas; cada salto añade latencia, costo y una frontera de fallo.

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

■ Flujo de razonamiento



Desarrollo

1. Cómo se elige una ruta

La causa número uno de confusión en redes es creer que la métrica o la preferencia deciden. No deciden mientras haya una ruta más específica.

```
ORDEN DE DECISIÓN

1. LONGITUD DE PREFIJO — el más específico gana
```

```
   destino 10.0.5.20
     10.0.0.0/8    → pasarela A
     10.0.0.0/16   → pasarela B
     10.0.5.0/24   → pasarela C     ← GANA

   y gana aunque A tenga mejor métrica, mejor preferencia
   o menos saltos

2. SOLO SI EMPATAN EN LONGITUD, se aplican preferencias
```

Y de aquí sale el ataque y el error más frecuentes:

```
anunciar un /24 dentro del /16 de otro DESVÍA su tráfico
→ es el mecanismo de los secuestros de ruta en internet
→ y también el de «alguien añadió una ruta a mano y ahora
  el tráfico va por el sitio equivocado»
```

Y el orden de desempate, simplificado a lo que se usa:

```
cuando dos rutas tienen la MISMA longitud
  1  origen: local o estática antes que aprendida
  2  preferencia local (política propia: por dónde salir)
  3  camino de AS más corto
  4  origen del anuncio, coste interno, y desempates finos
```

Y las dos rutas especiales que conviene tener claras:

```
0.0.0.0/0        por defecto; la menos específica de todas
                 → es la que se usa cuando nada más encaja

ruta a agujero negro
                 apunta a ninguna parte a propósito
                 → útil para bloquear un rango
                 → peligrosa si se propaga por error
```

Y una comprobación práctica que resuelve la mitad de los incidentes:

```
no preguntes «¿hay conectividad?»
pregunta «¿qué ruta coincide con este destino, en ESTA tabla?»
→ y luego la misma pregunta en la tabla de VUELTA
```

2. BGP: qué garantiza y qué no

BGP es el protocolo con el que dos redes se dicen qué prefijos alcanzan. Es lo que conecta internet y lo que conecta la nube con la red corporativa.

```
QUÉ HACE
  anunciar prefijos y aprender los del vecino
  aplicar POLÍTICA: qué anuncio, qué acepto, qué prefiero
  detectar que un vecino dejó de estar y retirar sus rutas

QUÉ NO HACE
  no mide latencia ni ancho de banda
  no elige el camino «más rápido»
  no valida que quien anuncia un prefijo sea su dueño
  no garantiza simetría
```

Y la consecuencia práctica más importante:

```
BGP es un protocolo de POLÍTICA, no de rendimiento
→ «el camino de AS más corto» no es el más rápido
→ y la mayoría de los problemas de latencia entre redes no
  se arreglan tocando BGP
```

Los conceptos que hacen falta, sin más:

```
AS                conjunto de redes con política común
                  → cada nube tiene el suyo; tu red
                    corporativa puede tener uno privado

SESIÓN            la conversación entre dos vecinos
                  → si cae, se retiran TODAS sus rutas

ANUNCIO           «yo alcanzo este prefijo»
RETIRADA          «ya no lo alcanzo»

PREFERENCIA LOCAL  qué salida prefiero YO   (entrante para mí)
```

```
PREPENDIENTE       repetir mi AS para parecer peor
                   → forma torpe pero habitual de influir en
                     cómo entra el tráfico
```

Y la asimetría del control, que es lo que más cuesta entender:

```
controlas bien POR DÓNDE SALES     (preferencia local)
controlas mal POR DÓNDE ENTRAS     (depende del otro)

→ para influir en la entrada solo puedes hacerte parecer peor
  o anunciar prefijos más específicos por el camino que
  prefieres
```

Lo que se rompe con BGP en la nube, que es casi siempre lo mismo:

```
LÍMITES DE PREFIJOS
  las pasarelas de nube aceptan un máximo (100, 200, 1.000)
  → si la corporativa anuncia 1.400 rutas, la sesión se CAE
  → y se cae entera, no parcialmente
  → solución: resumir en el origen                clase 193

ANUNCIO DEMASIADO AMPLIO
  anunciar 0.0.0.0/0 desde la corporativa hacia la nube
  → toda la salida a internet pasa por la corporativa
  → a veces se quiere; casi siempre es un accidente

LA SESIÓN SE RESTABLECE Y TARDA
  al recuperarse, reaprender miles de rutas lleva segundos
  o minutos, y durante ese tiempo hay agujeros
```

3. Propagación entre nube y corporativa

En la nube, las tablas de rutas son objetos que se asocian a subredes, y la propagación se activa o desactiva por decisión, no por protocolo.

```
LO QUE HAY QUE DECIDIR EXPLÍCITAMENTE
  qué prefijos anuncia la nube a la corporativa
  qué prefijos acepta la nube de la corporativa
  qué tablas reciben las rutas aprendidas
  qué subredes usan qué tabla
```

Y el error de diseño más común:

```
propagar todo a todas las tablas «para que funcione»
→ y entonces la subred de datos aprende la ruta a internet,
  o preproducción alcanza producción                clase 189
→ la propagación es un control de alcance, no una comodidad
```

Los patrones que funcionan:

```
AISLAMIENTO POR TABLA
  una tabla por función: pública, privada, datos
  → datos NO recibe la ruta por defecto a internet
  → y por tanto no puede sacar nada, aunque quiera

SALIDA CENTRALIZADA
  todo el tráfico saliente pasa por una red de inspección
  → ruta 0.0.0.0/0 hacia el cortafuegos central
  → permite registrar y controlar la salida     clase 200

RUTA MÁS ESPECÍFICA PARA EXCEPCIONES
  la excepción se implementa con un prefijo más largo
  → y por eso funciona sin tocar lo demás
```

La simetría, que rompe cosas de forma difícil de diagnosticar:

```
SI LA IDA Y LA VUELTA VAN POR CAMINOS DISTINTOS
  un cortafuegos con estado ve la respuesta sin haber visto
  la petición → la descarta
  el resultado: conexiones que se establecen y se cuelgan,
  o que funcionan en un sentido

CUÁNDO OCURRE
  dos túneles activos con políticas distintas en cada extremo
  una ruta más específica añadida en un solo sentido
  traducción de direcciones en un lado y no en el otro
```

```
  varias pasarelas NAT con reparto distinto
```

Y la comprobación:

```
traza el camino en los DOS sentidos, no en uno
→ y compara las tablas de rutas de ambos extremos
```

Y tres diagnósticos frecuentes con su firma:

```
SÍNTOMA                          CAUSA HABITUAL
funciona en un sentido           asimetría + cortafuegos con
                                 estado
funciona a ratos                 dos rutas iguales, reparto
                                 por flujo, una rota
deja de funcionar de golpe y     sesión BGP caída por exceso
todo a la vez                    de prefijos
alcanza un rango que no debería  propagación a una tabla que
                                 no tocaba
se pierde tráfico en silencio    ruta a agujero negro
                                 propagada           ley 13
```

4. Operar el encaminamiento

El encaminamiento es una de las pocas cosas cuyo fallo se manifiesta en todas partes a la vez, y por eso merece disciplina operativa propia.

```
LO QUE HAY QUE VIGILAR
  estado de cada sesión BGP, y su antigüedad        ley 13
  número de prefijos recibidos, contra el límite
  cambios en las rutas anunciadas y aprendidas
  aparición de rutas más específicas inesperadas
  asimetría en los caminos de ida y vuelta
```

Y la alerta que más incidentes evita:

```
«el número de prefijos recibidos está al 80 % del límite»
→ porque al 100 % la sesión se cae ENTERA y de golpe
→ y esa caída no avisa antes
```

Los cambios de encaminamiento, que merecen tratamiento especial:

```
UN CAMBIO DE RUTA AFECTA A CUANTO PASA POR AHÍ
  no hay despliegue escalonado natural
  → se hace en ventana, con vuelta atrás preparada
  → y con alguien mirando en los dos extremos

Y LA VUELTA ATRÁS TIENE QUE ESTAR PROBADA
  quitar una ruta es fácil; volver a poner el estado exacto
  anterior, no tanto
  → guardar la configuración previa, no describirla
```

Y una regla que se aprende cara:

```
las rutas añadidas «temporalmente» a mano no se quitan
→ y aparecen dos años después como causa de un incidente
→ toda ruta manual con fecha y dueño, o no se pone   ley 20
```

Lo que hay que documentar y casi nunca está:

```
qué prefijos anuncia cada extremo, y por qué
qué se acepta y qué se filtra
qué tabla usa cada subred
qué rutas son manuales, quién las puso y hasta cuándo
cuál es el camino esperado para los flujos principales
  → sin esto, nadie sabe si lo que ve es correcto
```

Y la lista de comprobación de la clase:

- ■ está documentado qué anuncia y qué acepta cada extremo
- ■ los anuncios están resumidos y por debajo del límite
- ■ hay alerta al 80 % del límite de prefijos
- ■ hay alerta de sesión caída y de antigüedad de la ruta
- ■ cada tabla de rutas tiene una función clara
- ■ la subred de datos no recibe ruta por defecto a internet
- ■ el camino de ida y el de vuelta se han trazado
- ■ no hay rutas manuales sin dueño ni fecha
- ■ los cambios de ruta tienen ventana y vuelta atrás guardada
- ■ está escrito el camino esperado de los flujos principales

Y el cierre que enlaza con la clase siguiente: saber por dónde va un paquete supone conocer la dirección de destino, y esa casi nunca se escribe: se resuelve por nombre. Cómo funciona esa resolución, y por qué es la primera sospechosa en tantos incidentes, es la materia de la clase 195.

Ejemplo trabajado

CloudShop conecta su red corporativa con tres nubes. Lo que sigue son tres incidentes de encaminamiento del mismo año, con su diagnóstico, y el rediseño que los cerró.

Incidente 1 · «Todo dejó de funcionar a la vez», marzo.

```
síntoma   a las 11:14, toda la conectividad entre la nube
          principal y la corporativa dejó de funcionar
          simultáneamente
          nada se había desplegado

primera hipótesis   corte del enlace dedicado
          → descartada: el enlace estaba activo

diagnóstico
  la sesión BGP hacia la pasarela de nube estaba caída
  motivo: prefijos recibidos 1.043, límite de la pasarela 1.000
  a las 11:14, el equipo de red corporativa había añadido una
  sede nueva → 61 prefijos más
  → se superó el límite y la sesión se cerró ENTERA

lo que lo hizo confuso
  el enlace físico estaba bien
  el panel de red decía «enlace activo»
  la sesión BGP no estaba vigilada

corrección inmediata   resumir los prefijos de las sedes
  1.043 rutas → 31 rutas resumidas

corrección de fondo
  alerta al 80 % del límite
  alerta de sesión caída, con 60 s de umbral
  resumen obligatorio en el origen                clase 193

tiempo hasta restablecer                          2 h 40
  de las cuales, hasta identificar la causa        2 h 05
```

Y la observación del análisis posterior:

```
el dato que resolvía el incidente —número de prefijos
recibidos— existía y se podía consultar en un comando
nadie lo miraba, y no había alerta                    ley 15
```

Incidente 2 · «Funciona en un sentido», junio.

```
síntoma   el servicio de informes en la nube podía consultar
          la base corporativa, pero las respuestas no llegaban
          las conexiones se establecían y se colgaban a los
          30 s (plazo del cliente)

diagnóstico
  había DOS caminos entre nube y corporativa
    el enlace dedicado principal
    un túnel de respaldo, activo
  la ida usaba el enlace dedicado (más específico desde la
    nube)
  la vuelta usaba el túnel (preferencia local en la
    corporativa)
  el cortafuegos corporativo, con estado, veía la respuesta
    sin haber visto la petición → la descartaba

causa raíz
  dos semanas antes, alguien había subido la preferencia
  local del túnel «para probar el respaldo» y no lo revirtió
  → ruta manual sin dueño ni fecha                    ley 20

corrección
  preferencia alineada en los dos extremos
```

```
  el túnel de respaldo pasa a estar en espera, no activo
  comprobación periódica de simetría en los 6 flujos
    principales
```

Incidente 3 · «Pérdida silenciosa hacia un rango», septiembre.

```
síntoma   un 4 % de las peticiones al servicio de inventario
          fallaban por plazo vencido; el resto funcionaba
          duró 11 días antes de detectarse

diagnóstico
  el inventario tiene 3 réplicas, en 10.0.132.0/24,
  10.0.133.0/24 y 10.0.134.0/24
  alguien había creado una ruta a agujero negro para
  10.0.134.0/24 durante una migración de agosto
  la ruta se propagó a la tabla de la subred de aplicación
  → un tercio del tráfico caía en el agujero
  → pero el reparto reintentaba, así que solo fallaba el
    4 % final

por qué tardó 11 días
  la tasa de error global era del 0,3 %, por debajo del
  umbral de alerta
  y el panel de disponibilidad seguía en verde       ley 13

corrección
  ruta retirada
  toda ruta a agujero negro exige etiqueta con dueño y fecha
  función de aptitud: ninguna ruta manual sin fecha de
    caducidad                                      clase 190
  alerta por tasa de error POR DESTINO, no solo global
```

El rediseño que salió de los tres incidentes.

```
TABLAS DE RUTAS POR FUNCIÓN
  pública      0.0.0.0/0 → pasarela de internet
  aplicación   0.0.0.0/0 → cortafuegos central
               10.64.0.0/12 → pasarela corporativa
  datos        SIN 0.0.0.0/0     ← no puede salir
               solo prefijos de aplicación y puntos privados
  puntos priv. solo lo necesario

efecto medido en el modelo de amenazas
  destinos alcanzables desde la subred de datos
    antes    cualquiera de internet
    después  4 prefijos internos
  → la prueba negativa de «sacar datos a un destino no
    declarado» pasó de fallar a rechazar          clase 189

ANUNCIOS Y FILTROS, DOCUMENTADOS
  la nube anuncia a la corporativa
    10.0.0.0/12   eu-west-1, resumido
    10.16.0.0/12  eu-central-1, resumido
    → 2 prefijos, no 340

  la nube ACEPTA de la corporativa
    10.64.0.0/12 y nada más
    → filtro explícito; 0.0.0.0/0 rechazado
    → antes se aceptaba todo, y en 2023 la corporativa
      anunció por error una ruta por defecto que desvió
      toda la salida a internet durante 40 minutos

VIGILANCIA
  sesión BGP: estado y antigüedad                   ley 13
  prefijos recibidos: alerta al 80 % del límite
  cambios en rutas anunciadas y aprendidas: alerta
  rutas manuales sin fecha: función de aptitud, 0 permitidas
  simetría de los 6 flujos principales: comprobación semanal
  tasa de error por destino, no solo global
```

El documento de caminos esperados, que no existía y resolvió el cuarto incidente en once minutos:

```
flujo                        camino esperado
app → base corporativa        enlace dedicado, ida y vuelta
app → internet                cortafuegos central → NAT
```

```
datos → puntos privados       sin salir de la red
informes → base corporativa   enlace dedicado, ida y vuelta
nube A → nube B               interconexión directa, no por
                              la corporativa
socios → nosotros             solo por el balanceador público
```

Y el resultado del año siguiente:

```
incidentes de encaminamiento                3           1
tiempo medio hasta identificar causa    1 h 50      11 min
prefijos anunciados a la corporativa       340           2
prefijos aceptados de la corporativa     1.043          31
rutas manuales sin dueño                    17           0
flujos con camino documentado                0           6
```

La lección que esta clase deja: los tres incidentes tuvieron causas distintas —un límite de prefijos, una preferencia que alguien cambió para probar, y una ruta a agujero negro de una migración— pero los tres se diagnosticaron lento por lo mismo: nadie había escrito cuál era el camino esperado, así que no había forma de saber si lo que se veía estaba mal. Y el tercero duró once días porque el error se repartía entre tres réplicas y quedaba por debajo del umbral global.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-16-advanced-cloud-networking-edge/194-routing-bgp-transito-y-propagacion-de-rutas/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 routing-lab 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 routing-lab 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 tráfico va por un camino que la métrica no explica	Existe una ruta más específica que gana antes de que se apliquen las preferencias	Busca siempre primero la ruta con prefijo más largo que coincide con el destino, en la tabla concreta que aplica.
Toda la conectividad cae a la vez sin que nadie despliegue nada	La sesión BGP superó el límite de prefijos y se cerró entera	Resume en el origen y alerta al 80 % del límite; vigila también estado y antigüedad de la sesión.
Las conexiones se establecen y se cuelgan, o funcionan en un solo sentido	Asimetría de camino con un cortafuegos con estado en medio	Traza ida y vuelta, alinea preferencias en ambos extremos y comprueba la simetría de los flujos principales periódicamente.
Una subred alcanza destinos que no debería	Se propagaron las rutas a todas las tablas para que funcionara	Usa una tabla por función y trata la propagación como control de alcance: la subred de datos no recibe ruta por defecto.
Se pierde una fracción del tráfico sin errores visibles	Una ruta a agujero negro propagada, con reintentos que enmascaran el fallo	Prohíbe rutas manuales sin dueño y fecha, y alerta por tasa de error por destino además de la global.
Nadie sabe si el camino que se observa es el correcto	No está escrito el camino esperado de los flujos principales	Documenta qué anuncia y acepta cada extremo, qué tabla usa cada subred y el camino esperado de cada flujo.

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 decide primero al elegir una ruta, y por qué las preferencias casi nunca importan?
2. ¿Qué garantiza BGP y qué no garantiza?
3. ¿Por qué se controla mal por dónde entra el tráfico?
4. ¿Qué ocurre cuando se supera el límite de prefijos de una pasarela?
5. ¿Qué documento hace que un incidente de encaminamiento se diagnostique en minutos?

Referencias

- RFC 4271 — A Border Gateway Protocol 4 (BGP-4). <https://www.rfc-editor.org/rfc/rfc4271>
- AWS (2025). Route tables and route priority in VPC. <https://docs.aws.amazon.com/vpc/latest/userguide/VpcSubnetRouting.html>
- Microsoft (2025). Virtual network traffic routing. <https://learn.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview>
- Google Cloud (2025). Routes and routing order. <https://cloud.google.com/vpc/docs/routes>
- Cloudflare (2025). What is BGP? Route leaks and hijacks. <https://www.cloudflare.com/learning/security/glossary/bgp-hijacking/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 16 en PDF · Manual integral

Anterior	Índice	Siguiente
← 193 · CIDR, subnetting y planificación IP a escala	Parte 16 · Programa	195 · DNS autoritativo, recursivo, split-horizon y DNSSEC →

Evaluación — 194 Routing, BGP, tránsito y propagación de rutas

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 routing-lab 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.

195 — DNS autoritativo, recursivo, split-horizon y DNSSEC

Parte: 16 — Redes cloud avanzadas, conectividad híbrida y edge Nivel: avanzado · Horas estimadas: 4 Laboratorio: network · Estado: EXECUTABLECORE

Propósito

Entender el sistema de nombres lo suficiente para diagnosticarlo, porque es la primera sospecha de casi todos los incidentes raros y la última que se comprueba bien. La clase separa autoritativo de recursivo, explica por qué el tiempo de vida es una decisión de arquitectura y no un ajuste, aborda la vista partida entre red interna y externa, y trata DNSSEC con honestidad: qué resuelve, qué no, y por qué su modo de fallo es apagar el dominio entero.

Resultados de aprendizaje

Al finalizar podrás:

1. Distinguir resolución autoritativa de recursiva y saber dónde mirar cada una.
2. Elegir tiempos de vida según lo que se vaya a cambiar y cuándo.
3. Diseñar vista partida sin que interno y externo se contradigan.
4. Evaluar si DNSSEC compensa, y operarlo sin apagar el dominio.
5. Diagnosticar los fallos de nombres por su firma característica.

Conceptos centrales

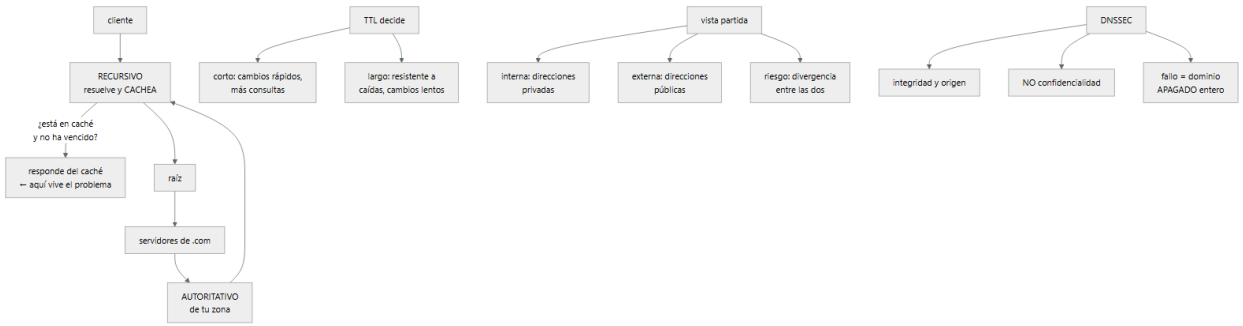
Concepto	Comprensión verificable
autoritativo	Servidor que posee la zona y responde con la verdad. Es donde se cambian los registros.
recursivo	Servidor que resuelve en nombre del cliente y guarda en caché. Es donde vive la respuesta que el cliente ve.
tiempo de vida (TTL)	Cuánto puede cachearse una respuesta. Determina cuánto tarda un cambio en surtir efecto en todas partes.
vista partida	La misma zona responde distinto según quién pregunte: red interna o internet.
DNSSEC	Firma criptográfica de las respuestas. Garantiza integridad y origen, no confidencialidad.
caché negativa	Almacenamiento de la respuesta «no existe». Su duración la fija un campo que casi nadie mira.

Modelo mental

La red cloud es un sistema distribuido de rutas, identidades y políticas; cada salto añade latencia, costo y una frontera de fallo.

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. Autoritativo y recursivo: dónde mirar

La mitad de los diagnósticos fallidos de nombres vienen de consultar el servidor equivocado.

```
AUTORITATIVO
  posee la zona; responde con la verdad actual
  es donde se CAMBIAN los registros
  no cachea nada: lo que dice es lo que hay

RECURSIVO
  resuelve en nombre del cliente y GUARDA la respuesta
  es lo que el cliente configura
  puede estar devolviendo algo viejo durante horas

Y EN MEDIO
  el sistema operativo cachea
  la biblioteca del lenguaje cachea      ← el peor
  el navegador cachea
  el proxy o el balanceador cachea
```

Y de ahí la regla de diagnóstico:

```
pregunta al AUTORITATIVO para saber qué hay configurado
pregunta al RECURSIVO que usa el cliente para saber qué ve
y compáralos

→ «yo lo veo bien» casi siempre significa «mi caché es otra»
```

Y una trampa concreta que aparece en producción:

```
algunas bibliotecas y máquinas virtuales de lenguaje cachean
la resolución PARA SIEMPRE dentro del proceso
→ un cambio de dirección no surte efecto hasta reiniciar
→ y el servicio parece «no ver» el cambio aunque el TTL
  haya vencido hace horas
→ comprobar el comportamiento de caché del entorno de
  ejecución es parte del diseño, no del diagnóstico
```

Los tipos de registro que hacen falta, sin más:

```
A / AAAA     nombre → dirección IPv4 / IPv6
CNAME        nombre → otro nombre
             ojo: no puede coexistir con otros registros del
             mismo nombre, y no vale en la raíz del dominio
ALIAS/ANAME  registro propietario que resuelve eso último
NS           quién es autoritativo para la zona
MX           correo
TXT          verificaciones, políticas de correo
SRV          servicio, puerto y prioridad
CAA          qué autoridades pueden emitir certificados
             ← el que casi nadie pone y evita emisiones
               no autorizadas
PTR          dirección → nombre; usado por sistemas de correo
```

2. El tiempo de vida es una decisión de arquitectura

El TTL se rellena por costumbre y determina cosas importantes.

```
TTL CORTO   (30-60 s)
```

```
  + un cambio surte efecto casi enseguida
  + permite conmutación por nombre              clase 166
  − más consultas y más dependencia del recursivo
  − si el autoritativo cae, todo expira rápido y se apaga

TTL LARGO   (1-24 h)
  + resistente: si el autoritativo cae, todo sigue
  + menos consultas
  − un cambio tarda hasta un día en verse en todas partes
  − y en una emergencia, ese día es el tiempo de recuperación
```

Y la regla que resuelve la elección:

```
el TTL debe ser MENOR que el tiempo de recuperación que
prometes por ese nombre

→ si prometes conmutar en 15 minutos y el TTL es de 1 hora,
  no cumples, hagas lo que hagas               clase 185
```

Y el procedimiento para cambiar algo que tiene TTL largo:

```
1  bajar el TTL a 60 s
2  ESPERAR el TTL antiguo entero (hasta 24 h)
3  hacer el cambio
4  comprobar
5  volver a subir el TTL

→ saltarse el paso 2 es el error clásico: parte del mundo
  sigue con la respuesta vieja
```

Y dos valores que casi nadie mira y causan incidentes:

```
CACHÉ NEGATIVA
  cuánto se recuerda que un nombre NO existe
  se fija en el registro SOA de la zona, no por registro
  → si vale 3 h y creas un nombre nuevo tras haberlo
    consultado, tardará 3 h en verse
  → causa típica de «he creado el registro y no funciona»

TTL DE LOS REGISTROS NS
  cuánto se recuerda quién es autoritativo
  → un cambio de proveedor de DNS tarda ese tiempo, y suele
    ser de 24-48 h
```

Y una decisión de diseño que evita mucho dolor:

```
no uses nombres con TTL largo para lo que va a cambiar
separa los nombres estables de los que conmutan
  api.cloudshop.com          TTL 60   ← conmuta
  docs.cloudshop.com         TTL 3600 ← estable
```

3. Vista partida y zonas internas

Casi toda organización acaba resolviendo el mismo nombre de dos formas: por dentro apunta a una dirección privada, por fuera a una pública.

```
POR QUÉ SE HACE
  el tráfico interno no debe salir a internet para volver
  → latencia, coste de salida y exposición    clase 200
  y el certificado y el nombre siguen siendo los mismos

CÓMO SE HACE
  una zona privada, asociada a las redes internas
  con los mismos nombres y direcciones privadas
  → el resolutor interno la consulta primero
```

Y los tres problemas que trae, con su solución:

```
1. DIVERGENCIA
   alguien añade un registro en la zona pública y no en la
   privada, o al revés
   → los internos ven una cosa y los externos otra
   solución   generar ambas del mismo origen, o comprobar
              la diferencia automáticamente        clase 190

2. NOMBRES QUE SOLO EXISTEN DENTRO
```

```
   y que se filtran en registros, correos o mensajes de error
   → revelan estructura interna                    clase 189

3. DIAGNÓSTICO CONFUSO
   «a mí me resuelve bien» según desde dónde se pregunte
   solución   decir SIEMPRE desde qué red se consultó
```

La resolución en la nube, que tiene sus propias reglas:

```
cada red tiene un resolutor propio, en una dirección fija
  del rango de la red
las zonas privadas se ASOCIAN a redes concretas
  → una red sin asociar no ve la zona
la resolución desde la corporativa hacia la nube exige
  reenvío explícito, y viceversa
  → punto de entrada y de salida del resolutor
```

Y el fallo más común de este montaje:

```
el reenvío está configurado en un sentido y no en el otro
→ desde la nube se resuelven los nombres corporativos
→ desde la corporativa no se resuelven los de la nube
→ y se descubre el día que un sistema corporativo tiene que
  llamar a un servicio nuevo
```

Y una decisión que ahorra trabajo:

```
usa un subdominio propio para lo interno
  int.cloudshop.com     para todo lo privado
→ evita la vista partida en la mayoría de los nombres
→ y deja la partida solo donde de verdad hace falta
```

4. DNSSEC y los fallos característicos

DNSSEC firma las respuestas para que el resolutor pueda comprobar que no han sido alteradas.

```
QUÉ RESUELVE
  que alguien altere la respuesta por el camino
  que un resolutor sea envenenado con datos falsos

QUÉ NO RESUELVE
  confidencialidad: la consulta viaja en claro igual
  → eso lo resuelven DNS sobre TLS o sobre HTTPS
  que el dominio esté mal configurado
  que alguien tome el control de tu cuenta de DNS
```

Y el motivo por el que se adopta despacio:

```
SU MODO DE FALLO ES APAGAR EL DOMINIO ENTERO
  si una firma caduca o la cadena se rompe, los resolutores
  que validan RECHAZAN todas las respuestas
  → no es degradación: es desaparición
  → y el diagnóstico desde fuera parece «el dominio no existe»
```

Y por eso su operación tiene reglas propias:

```
vigilar la caducidad de las firmas, con margen  ← ley 13
vigilar que el registro de delegación en el dominio padre
  coincide con la clave activa
cualquier cambio de proveedor de DNS exige un procedimiento
  específico, y es donde más se rompe
y la rotación de claves se ensaya antes                ley 22
```

Los fallos de nombres por su firma, que es lo que permite diagnosticar rápido:

SÍNTOMA	CAUSA HABITUAL
funciona en unos sitios y no en otros	caché con TTL vigente en unos recursivos
funciona y de pronto deja de funcionar, y luego vuelve	registro cambiado; unos recursivos ya expiraron y otros no
creo un registro y no aparece	caché negativa del SOA
el servicio no ve la dirección nueva tras horas	la biblioteca cachea dentro del proceso

```
resuelve distinto según el equipo    vista partida y red
                                     distinta

el dominio entero desaparece         DNSSEC roto, o los NS
                                     apuntan a un proveedor
                                     que ya no sirve la zona

latencia alta al conectar            resolución lenta o
solo la primera vez                  resolutor sin caché

resuelve pero con dirección de       registro huérfano de un
un servicio retirado                 recurso ya borrado
                                     ← peligro real
```

Y el último merece su propia advertencia:

```
REGISTRO HUÉRFANO
  un nombre que apunta a una dirección o a un recurso que ya
  no es tuyo
  → alguien puede reclamar ese recurso y recibir tu tráfico
  → afecta a subdominios apuntando a servicios externos
    dados de baja
  → comprobación periódica obligatoria             ley 20
```

Y la lista de comprobación de la clase:

- se sabe qué es autoritativo y qué recursivo en cada caso
- los TTL son menores que el plazo de recuperación prometido
- los nombres que conmutan están separados de los estables
- el TTL de caché negativa está revisado
- se conoce el comportamiento de caché del entorno de ejecución
- la vista partida se genera del mismo origen o se compara
- el reenvío entre nube y corporativa funciona en ambos sentidos
- hay registro CAA
- si hay DNSSEC, hay alerta de caducidad de firmas
- se comprueban periódicamente los registros huérfanos
- hay alerta si la zona deja de responder o cambia sola

Y el cierre que enlaza con la clase siguiente: resuelto el nombre, el tráfico llega a un punto de entrada que lo reparte y termina la conexión cifrada. Balanceo, proxies y gestión de certificados es la materia de la clase 196.

Ejemplo trabajado

CloudShop sufre cuatro incidentes de nombres en un año. Lo que sigue es cada uno con su firma, el rediseño que salió, y el registro huérfano que encontraron por casualidad.

Incidente 1 · «El cambio no surte efecto», enero.

```
contexto  migración del balanceador de pedidos a uno nuevo
          se cambió el registro A de api.cloudshop.com

síntoma   a las 2 h, el 30 % del tráfico seguía llegando al
          balanceador viejo, que ya estaba apagado a medias

diagnóstico
  el TTL del registro era 86.400 (24 h)
  el equipo lo bajó a 60 s el mismo día del cambio
  → los recursivos que ya tenían la respuesta antigua la
    mantuvieron hasta 24 h

qué faltó   bajar el TTL y ESPERAR el TTL antiguo entero
            ANTES de cambiar

efecto      6 h de tráfico partido; 340 pedidos fallidos
```

Incidente 2 · «He creado el registro y no aparece», marzo.

```
contexto  se creó pagos-v2.cloudshop.com para una prueba

síntoma   el equipo lo consultó antes de crearlo (por error),
          y después de crearlo seguía sin resolver durante
          casi tres horas, solo desde algunas redes
```

```
diagnóstico
  caché negativa: el campo mínimo del SOA valía 10.800 (3 h)
  el resolutor había guardado «este nombre no existe»

corrección
  caché negativa bajada a 300 s
  y una nota en el procedimiento: no consultar un nombre
  antes de crearlo
```

Incidente 3 · «Resuelve bien desde la nube y mal desde la oficina», junio.

```
síntoma   el sistema corporativo de conciliación no podía
          llamar al servicio de inventario nuevo
          desde la nube todo funcionaba

diagnóstico
  zona privada int.cloudshop.com, asociada a las redes de
  la nube
  el reenvío desde la corporativa hacia el resolutor de la
  nube NO existía
  → el reenvío estaba montado solo en el sentido
    nube → corporativa

corrección
  punto de entrada del resolutor en la nube y regla de
  reenvío en la corporativa
  comprobación negativa añadida: resolver un nombre interno
  desde CADA red, en las dos direcciones
```

Incidente 4 · «El servicio no ve la dirección nueva», septiembre.

```
contexto  conmutación de la base de datos a la réplica de
          otra zona; el nombre se actualizó y el TTL era 30 s

síntoma   4 de los 11 servicios siguieron conectando a la
          dirección antigua durante 40 minutos, hasta que
          se reiniciaron

diagnóstico
  los 4 servicios corrían sobre un entorno de ejecución que
  cachea la resolución dentro del proceso, sin respetar el
  TTL
  → el TTL de 30 s no servía de nada para esos 4

corrección
  configuración explícita de caducidad de caché en el
  entorno de ejecución
  y una prueba negativa nueva: cambiar el nombre y comprobar
  que CADA servicio lo sigue en menos de 60 s

y el hallazgo de método
  el plazo de recuperación prometido para la base era de
  5 minutos
  el TTL era correcto, y aun así no se cumplía
  → el plazo depende de la capa que más cachea, no del TTL
```

El rediseño de nombres.

```
SEPARACIÓN POR VOLATILIDAD
  nombres que conmutan          TTL 30 s
    api, base de datos, balanceadores
  nombres estables              TTL 3.600 s
    documentación, sitio corporativo
  nombres de infraestructura    TTL 300 s

SUBDOMINIO INTERNO
  todo lo privado en int.cloudshop.com
  → la vista partida queda solo en 3 nombres, no en 60

GENERACIÓN ÚNICA
  las zonas pública e interna se generan del mismo origen
  declarativo
  → función de aptitud: ningún registro existe en una zona
    y no en la otra sin declararlo             clase 190
```

```
CACHÉ NEGATIVA          300 s
REGISTRO CAA            añadido; solo dos autoridades
DNSSEC                  evaluado y NO adoptado, ver abajo
```

La decisión sobre DNSSEC, registrada:

```
CONTEXTO
  el dominio no maneja correo sensible
  el tráfico crítico va sobre TLS con validación de
  certificado, que ya detecta la suplantación
  el equipo tiene 4 personas y no hay guardia nocturna

DECISIÓN   no adoptar DNSSEC en esta fase

MOTIVO
  su modo de fallo apaga el dominio entero, y el equipo no
  tiene capacidad de vigilar caducidades de firma con la
  disciplina que exige
  el riesgo que mitiga está ya cubierto en su mayor parte
  por TLS para el tráfico que importa

QUÉ LA REABRIRÍA
  si aparece un requisito contractual
  si se empieza a manejar correo con datos sensibles
  si el proveedor pasa a gestionar la rotación de claves
  automáticamente y con alerta de caducidad

ACEPTA     dirección de tecnología, por escrito
REVISA     12 meses
```

Y la observación del registro:

```
esta es una decisión de NO adoptar una buena práctica
→ y por eso lleva motivo, nombre y fecha            clase 189
→ «no lo hicimos» sin registro es un olvido; con registro
  es una decisión
```

El registro huérfano, encontrado al inventariar:

```
se comprobaron los 214 registros de la zona pública

registros que apuntaban a recursos inexistentes            9
  de ellos, a direcciones que ya no eran de CloudShop      3
  de ellos, a un servicio externo dado de baja en 2023     1

el peligroso
  encuestas.cloudshop.com apuntaba por CNAME a una
  plataforma de formularios cuya cuenta se canceló en 2023
  → cualquiera que registrase ese identificador en esa
    plataforma habría servido contenido bajo el dominio de
    CloudShop
  → llevaba 19 meses así

corrección
  registros huérfanos eliminados
  comprobación mensual automatizada: todo CNAME y todo A
  debe apuntar a algo que existe y es nuestro
  → 3 más detectados en los 6 meses siguientes
```

El resultado del año siguiente:

```
incidentes de nombres                          4          1
tiempo medio de diagnóstico                55 min     9 min
nombres con vista partida                     60         3
registros huérfanos                            9         0
tiempo real de conmutación por nombre      40 min     50 s
divergencias entre zona interna y pública      7         0
```

La lección que esta clase deja: de los cuatro incidentes, tres tuvieron como causa una caché que nadie había considerado —la de los recursivos con el TTL antiguo, la negativa del SOA y la del entorno de ejecución— y ninguna estaba en el diagrama de arquitectura. Y el hallazgo más grave del año no fue un incidente: fue un CNAME olvidado durante diecinueve meses que apuntaba a una cuenta cancelada, y lo encontró un inventario, no una alerta.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-16-advanced-cloud-networking-edge/195-dns-autoritativo-recursivo-split-horizon-y-dnssec/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 dns-design 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 dns-design 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 cambio de registro tarda horas en surtir efecto para parte del tráfico	Se bajó el TTL el mismo día del cambio, sin esperar a que expirara el antiguo	Baja el TTL, espera el TTL antiguo completo, haz el cambio, comprueba y vuelve a subirlo.
Se crea un registro y no resuelve durante horas	Caché negativa: el nombre se consultó antes de existir y se recordó como inexistente	Revisa el valor mínimo del SOA y bájalo; evita consultar nombres antes de crearlos.
Un servicio sigue conectando a la dirección antigua pese al TTL corto	La biblioteca o el entorno de ejecución cachea la resolución dentro del proceso	Configura explícitamente la caducidad de caché del entorno y añade una prueba negativa que cambie el nombre y mida cuánto tarda cada servicio en seguirlo.
Un nombre interno resuelve desde la nube y no desde la red corporativa	El reenvío entre resolutores está configurado en un solo sentido	Monta punto de entrada y de salida del resolutor, y comprueba la resolución desde cada red en ambas direcciones.
Internos y externos ven direcciones distintas y contradictorias	Las zonas pública y privada se mantienen por separado y divergen	Genera ambas del mismo origen declarativo y comprueba automáticamente que no haya registros en una y no en la otra.
Un subdominio sirve contenido que no es tuyo	Registro huérfano apuntando a un recurso o cuenta externa dada de baja	Inventaría periódicamente todos los registros y comprueba que cada destino existe y es tuyo.

■ 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. ¿A qué servidor se pregunta para saber qué hay configurado y a cuál para saber qué ve el cliente?
2. ¿Qué relación debe cumplir el TTL con el plazo de recuperación prometido?
3. ¿Qué campo fija la caché negativa y qué síntoma produce?
4. ¿Cuál es el modo de fallo de DNSSEC y por qué condiciona su adopción?
5. ¿Qué es un registro huérfano y por qué es peligroso?

Referencias

- RFC 1034 y 1035 — Domain names: concepts, facilities and implementation. <https://www.rfc-editor.org/rfc/rfc1034>
- RFC 2308 — Negative caching of DNS queries. <https://www.rfc-editor.org/rfc/rfc2308>
- RFC 4033 — DNS security introduction and requirements (DNSSEC). <https://www.rfc-editor.org/rfc/rfc4033>
- AWS (2025). Route 53: private hosted zones and resolver endpoints. <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/hosted-zones-private.html>
- OWASP (2025). Subdomain takeover — registros huérfanos. <https://owasp.org/www-community/attacks/SubdomainTakeover>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 16 en PDF · Manual integral

Anterior	Índice	Siguiente
← 194 · Routing, BGP, tránsito y propagación de rutas	Parte 16 · Programa	196 · Balanceo L4/L7, proxies, TLS y gestión de certificados →

Evaluación — 195 DNS autoritativo, recursivo, split-horizon y DNSSEC

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 dns-design 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.

196 — Balanceo L4/L7, proxies, TLS y gestión de certificados

Parte: 16 — Redes cloud avanzadas, conectividad híbrida y edge Nivel: avanzado · Horas estimadas: 4 Laboratorio: network · Estado: EXECUTABLECORE

Propósito

Elegir y operar el punto de entrada del tráfico: qué reparte, en qué capa, con qué algoritmo, y cómo termina el cifrado. La clase separa balanceo de capa 4 y de capa 7 por lo que cada uno puede y no puede hacer, revisa los algoritmos de reparto con la evidencia de la clase 186 sobre por qué el rotatorio es mala idea, y trata la gestión de certificados como lo que en la práctica es: la causa recurrente de caídas totales y previsibles.

Resultados de aprendizaje

Al finalizar podrás:

1. Elegir entre balanceo de capa 4 y de capa 7 según lo que haga falta.
2. Seleccionar el algoritmo de reparto y las comprobaciones de salud.
3. Decidir dónde termina TLS y qué se cifra por dentro.
4. Automatizar la emisión y renovación de certificados sin sorpresas.
5. Diagnosticar los fallos típicos de balanceador y de certificado.

Conceptos centrales

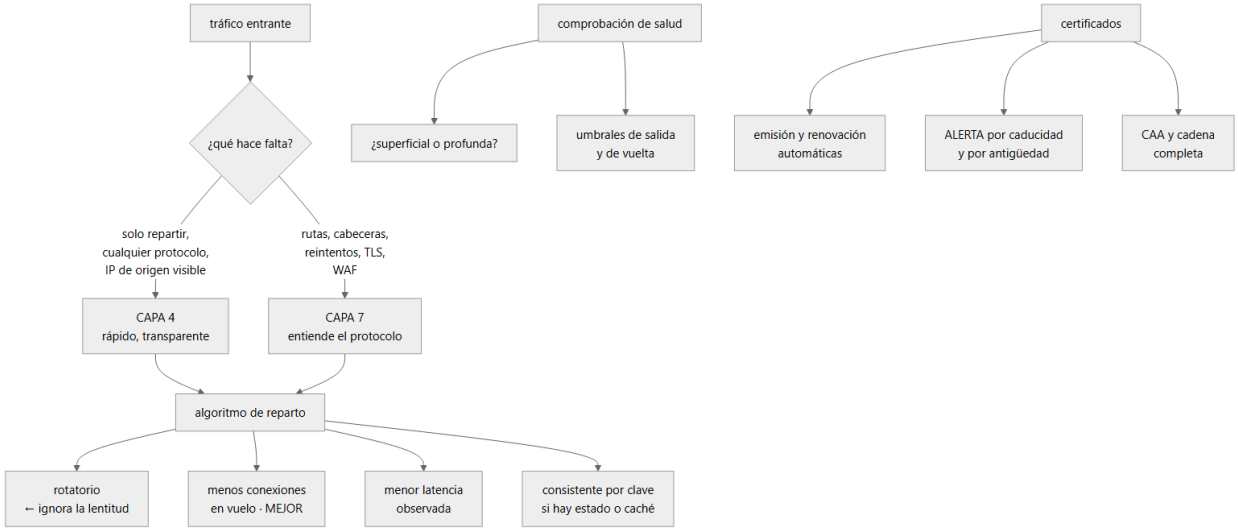
Concepto	Comprensión verificable
capa 4	Reparto por dirección y puerto, sin ver el contenido. Rápido, transparente y ciego.
capa 7	Reparto entendiendo el protocolo: rutas, cabeceras, reintentos. Más capaz y más caro.
comprobación de salud	Sonda periódica que decide si un destino recibe tráfico. Determina cuánto tarda en salir uno roto.
drenaje de conexiones	Dejar terminar lo que está en curso antes de retirar un destino.
terminación TLS	Punto donde se descifra el tráfico. Decide qué ve cada capa y qué hay que cifrar por dentro.
SNI	Extensión que indica el nombre solicitado antes de descifrar. Permite varios certificados en una IP.

Modelo mental

La red cloud es un sistema distribuido de rutas, identidades y políticas; cada salto añade latencia, costo y una frontera de fallo.

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. Capa 4 y capa 7

La elección no es de gusto: cada capa puede cosas que la otra no.

```
CAPA 4  (transporte)
  ve   dirección y puerto
  no ve  rutas, cabeceras, cuerpo
  hace   reparte conexiones a destinos

  a favor
    latencia mínima; apenas añade nada
    funciona con cualquier protocolo, no solo HTTP
    puede preservar la dirección de origen del cliente
    escala a millones de conexiones
  en contra
    no puede reintentar una petición fallida
    no distingue una ruta de otra
    no termina TLS (salvo modos específicos)
    su comprobación de salud es superficial

CAPA 7  (aplicación)
  ve   ruta, cabeceras, método, cuerpo
  hace   enruta por contenido, reintenta, reescribe,
         termina TLS, aplica límites y filtros

  a favor
    reparto por ruta o cabecera → despliegues escalonados
    reintento de peticiones idempotentes
    plazos por ruta
    terminación TLS y certificados centralizados
    filtrado de aplicación
  en contra
    añade latencia (0,3-2 ms típicos)
    es un punto que hay que dimensionar y operar
    ve el tráfico en claro → decisión de seguridad
```

Y la regla de elección:

```
¿necesitas decidir según el CONTENIDO de la petición?
  sí   capa 7
  no   capa 4, y te ahorras latencia y complejidad

¿el protocolo no es HTTP?
  capa 4, salvo que el producto entienda ese protocolo
```

Y la combinación habitual, que resuelve casi todo:

```
internet → capa 4 (entrada, IP fija, escala)
         → capa 7 (rutas, TLS, reintentos)
         → servicios
```

```
→ y la IP fija del capa 4 es la que se pone en las listas
  de permitidos de terceros                          clase 193
```

2. Reparto y comprobaciones de salud

El algoritmo de reparto por defecto suele ser el rotatorio, y suele ser el peor.

```
ROTATORIO
  una conexión a cada destino, por turno
  problema   ignora que un destino esté lento
  → una instancia degradada recibe la misma carga que una
    sana, y arruina el percentil de todos      clases 152, 186

MENOS CONEXIONES EN VUELO
  al destino con menos peticiones activas
  → un destino lento acumula peticiones y deja de recibir
  → es el que mejor se comporta ante degradación
  → suele ser la elección correcta por defecto

MENOR LATENCIA OBSERVADA
  usa la latencia medida reciente
  → bueno, pero puede oscilar si la ventana es corta

CONSISTENTE POR CLAVE (hash)
  la misma clave va siempre al mismo destino
  → necesario con caché local o estado por sesión
  → usar variante con anillo para que al perder un destino
    solo se reubique su parte, no todo

PESOS
  para instancias de tamaño distinto o para desplazar
  tráfico progresivamente                            clase 102
```

Las comprobaciones de salud, que deciden cuánto tarda en salir un destino roto:

```
SUPERFICIAL   «¿responde el puerto?» o «¿devuelve 200 en /?»
  ventaja   barata
  problema  un proceso vivo con la base caída pasa la prueba

PROFUNDA      comprueba las dependencias críticas
  ventaja   detecta el destino inútil
  problema  si la dependencia común cae, TODOS fallan la
            comprobación y el servicio desaparece entero

LA PRÁCTICA QUE FUNCIONA
  dos puntos distintos
    /vivo    ¿el proceso está vivo?     → reinicio
    /listo   ¿puede atender ahora?      → reparto
  y /listo NO debe depender de dependencias blandas
```

Y los parámetros que importan más de lo que parece:

```
intervalo × fallos para salir = tiempo hasta dejar de recibir
  10 s × 3 = 30 s de tráfico enviado a un destino roto

y para volver, más estricto que para salir
  2 fallos para salir, 5 aciertos para volver
  → evita el destino que entra y sale sin parar
```

Y dos comportamientos que hay que configurar siempre:

```
DRENAJE DE CONEXIONES
  al retirar un destino, dejar terminar lo que está en curso
  → sin esto, cada despliegue corta peticiones vivas
  → y el plazo de drenaje debe superar el plazo más largo
    de las peticiones                                clase 186

EXPULSIÓN DE ATÍPICOS
  retirar temporalmente un destino que devuelve errores,
  aunque su comprobación de salud pase
  → cubre el fallo gris que la sonda no ve       clase 185
```

3. TLS: dónde termina y qué se cifra dentro

Decidir dónde se descifra el tráfico es una decisión de arquitectura con consecuencias de seguridad, latencia y diagnóstico.

```
TERMINACIÓN EN EL BORDE
  el balanceador descifra; hacia dentro va en claro
  + certificados en un sitio; menos carga; se puede enrutar
    por contenido
  − el tráfico interno viaja sin cifrar

TERMINACIÓN EN EL BORDE Y RECIFRADO
  descifra, decide, y vuelve a cifrar hacia el destino
  + lo mejor de los dos, y es lo habitual hoy
  − doble coste de cifrado

PASO DIRECTO
  el balanceador no descifra; el destino termina TLS
  + confidencialidad extremo a extremo
  − no se puede enrutar por ruta ni cabecera; el certificado
    vive en cada destino

TLS MUTUO
  ambos extremos presentan certificado
  → identidad de carga sin credenciales           clase 201
```

Y la decisión práctica:

```
en la nube, terminar en el borde y RECIFRAR hacia dentro es
la opción por defecto razonable
→ y el cifrado interno deja de ser opcional en cuanto el
  tráfico cruza una frontera de confianza          clase 189
```

Los certificados, que causan caídas totales con una regularidad notable:

```
POR QUÉ CAEN
  caducan; el aviso llegó a un buzón que nadie lee   ley 15
  la renovación automática falla en silencio         ley 13
  falta un eslabón de la cadena y unos clientes fallan
    y otros no
  el certificado no cubre el nombre exacto solicitado
  se renovó en un sitio y no en los otros tres
```

Y las reglas que evitan casi todas:

```
1  EMISIÓN Y RENOVACIÓN AUTOMÁTICAS
   sin intervención humana; con protocolo estándar

2  ALERTA POR ANTIGÜEDAD, NO SOLO POR CADUCIDAD    ley 13
   «este certificado no se ha renovado en X días»
   → detecta la automatización rota ANTES de caducar

3  ALERTA A 30, 14 Y 7 DÍAS, a un canal con guardia

4  INVENTARIO DE TODOS los certificados
   incluidos los de dispositivos, agentes y llamadas
   salientes
   → los que no están en el balanceador son los que caducan

5  CADENA COMPLETA SERVIDA
   comprobado desde un cliente sin almacén completo

6  REGISTRO CAA en el dominio                    clase 195

7  PRUEBA NEGATIVA: adelantar el reloj o usar un
   certificado caducado y comprobar que la alerta salta
                                                   ley 22
```

Y el caso que más se olvida:

```
los certificados de CLIENTE que usamos para llamar a
terceros
→ no están en ningún balanceador
→ nadie los vigila
→ y su caducidad rompe una integración concreta, no la web
```

4. Diagnóstico y operación

Los fallos de balanceador tienen firmas reconocibles, y saberlas ahorra horas.

SÍNTOMA	CAUSA HABITUAL
errores 502 intermitentes	destino que cierra la conexión antes que el balanceador; plazos mal alineados
errores 504	el destino no responde a tiempo; plazo del balanceador menor que el trabajo real
todos los destinos «no sanos» a la vez	la comprobación es profunda y cayó la dependencia común
un destino recibe más carga que los demás	reparto consistente por clave con clave sesgada
el p99 empeora y el p50 no	un destino degradado que sigue recibiendo (rotatorio)
cada despliegue corta peticiones	sin drenaje de conexiones
funciona en unos clientes y no en otros	cadena de certificado incompleta
falla solo el primer intento tras escalar	el destino nuevo aún no pasó la comprobación

Y una regla de coherencia que evita los dos primeros:

```
plazo del cliente > plazo del balanceador > plazo del destino
  y el de conexión inactiva del destino MAYOR que el del
  balanceador
→ si el destino cierra antes, el balanceador reutiliza una
  conexión muerta y devuelve 502
```

Lo que hay que vigilar en el punto de entrada:

```
peticiones por segundo y por código de estado
latencia por percentil, medida AQUÍ (es el borde) clase 126
destinos sanos frente a configurados               ley 13
conexiones activas y en cola
reintentos y expulsiones de atípicos
certificados: días hasta caducar y días desde la última
  renovación
```

Y la lista de comprobación de la clase:

- ■ la capa elegida corresponde a lo que hay que decidir
- ■ el algoritmo no es rotatorio en servicios con latencia variable
- ■ hay puntos separados de vivo y de listo
- ■ /listo no depende de dependencias blandas
- ■ el tiempo hasta retirar un destino roto está calculado
- ■ volver es más estricto que salir
- ■ hay drenaje de conexiones, mayor que el plazo más largo
- ■ hay expulsión de atípicos para el fallo gris
- ■ los plazos están ordenados cliente > balanceador > destino
- ■ TLS termina donde se decidió, y hacia dentro va cifrado
- ■ la emisión y renovación de certificados son automáticas
- ■ hay alerta por antigüedad de renovación, no solo por caducidad
- ■ el inventario incluye los certificados de cliente
- ■ la cadena completa se sirve y se ha comprobado

Y el cierre que enlaza con la clase siguiente: el punto de entrada reparte lo que le llega, pero lo más barato es que buena parte del tráfico no llegue nunca hasta él. Distribución de contenido, caché en el borde y cómputo en el borde es la materia de la clase 197.

Ejemplo trabajado

CloudShop revisa su punto de entrada tras dos caídas totales por certificado y un problema de percentiles que llevaba meses. Lo que sigue es el diagnóstico de los tres, y el rediseño con sus cifras.

Caída total 1 · certificado caducado, febrero.

```
síntoma   a las 03:12, todo el tráfico público empezó a
          fallar con error de certificado
          duración: 1 h 47

diagnóstico
  el certificado comodín *.cloudshop.com caducó
  la renovación automática existía y llevaba 71 días
  fallando en silencio
  motivo del fallo: la validación por DNS requería un
  registro TXT que se borró al reorganizar la zona en
  noviembre

por qué nadie lo supo
  la alerta de caducidad estaba configurada a 7 días
  y salía a un canal de correo que se creó en 2022 y no
  tenía suscriptores                        ley 15, clase 194

coste     1 h 47 de caída total; 2.100 pedidos perdidos
```

Caída total 2 · el certificado renovado en un sitio, mayo.

```
síntoma   el 30 % de las peticiones fallaban por certificado,
          y el resto no

diagnóstico
  había 4 puntos de entrada: 2 balanceadores públicos, una
  distribución de contenido y una pasarela de API
  la renovación automática cubría los 2 balanceadores
  la pasarela de API tenía el certificado subido a mano en
  2023
  → caducó, y el 30 % del tráfico pasaba por ahí

el hallazgo del inventario posterior
  certificados en uso              31
  bajo renovación automática       12
  subidos a mano                   19
  de ellos, de CLIENTE (para llamar a terceros)     6
  → los 6 no estaban en ningún inventario
```

El problema de percentiles, diagnosticado en junio.

```
síntoma   desde hacía meses, el p99 del listado era de
          2.100 ms mientras el p50 estaba en 70 ms
          se atribuía a «consultas pesadas»

diagnóstico
  el algoritmo de reparto era rotatorio
  una de las 12 instancias tenía un disco degradado y
  respondía 40 veces más lento
  su comprobación de salud (200 en /) pasaba perfectamente
  → recibía 1 de cada 12 peticiones y arruinaba el p99

comprobación
  al retirar esa instancia a mano, el p99 bajó a 240 ms
  en 4 minutos

y cuánto llevaba así
  se revisó el histórico: el patrón aparecía desde marzo
  → 3 meses                                      ley 13
```

El rediseño del punto de entrada.

```
ARQUITECTURA
  internet
    → balanceador de capa 4, IP fija     ← lista de permitidos
      de socios
      → balanceador de capa 7
```

```
        · terminación TLS y recifrado hacia dentro
        · enrutado por ruta (api, web, estáticos)
        · reintento de GET idempotentes, máximo 1
        · plazo por ruta
        → servicios

motivo de la capa 4 delante
  las direcciones públicas quedan fijas y pocas
  → renumerar el capa 7 no obliga a tocar listas de terceros
                                                  clase 193

REPARTO Y SALUD
  algoritmo        menos conexiones en vuelo
                   (antes: rotatorio)
  /vivo            proceso arriba
  /listo           puede atender: comprueba la base y el
                   caché propio, NO el servicio de
                   recomendaciones (blanda)
  salir            2 fallos × 5 s = 10 s
  volver           5 aciertos × 5 s = 25 s
  drenaje          45 s (plazo más largo: 30 s)
  expulsión de     5xx > 5 % en 30 s → fuera 30 s
  atípicos         ← esto es lo que cubre el fallo gris
```

Y el efecto del cambio de algoritmo, medido:

```
p50 del listado                          70 ms      68 ms
p99 del listado                       2.100 ms     235 ms
destinos degradados detectados             0/mes    1,3/mes
tiempo hasta retirar uno degradado       manual       35 s
```

Y la observación importante:

```
el cambio no fue de capacidad ni de código
fue dejar de repartir por turnos y añadir expulsión de
atípicos
→ el problema llevaba 3 meses y costaba 0 € arreglarlo

CERTIFICADOS
  inventario completo, generado del estado real         31
  bajo renovación automática                            31
    → incluidos los 6 de cliente hacia terceros

  alertas
    caducidad a 30, 14 y 7 días → canal de guardia
    ANTIGÜEDAD: «sin renovar desde hace más de 45 días»
      → esta es la que habría detectado el fallo de febrero
        26 días antes de la caída
    validación de la cadena completa, semanal, desde un
      cliente con almacén mínimo

  registro CAA con dos autoridades                clase 195

  prueba negativa trimestral
    romper a propósito la validación de un certificado de
    prueba y comprobar que la alerta de antigüedad salta
    → primera ejecución: la alerta salió a un canal
      equivocado; corregido                          ley 22

PLAZOS, ORDENADOS
  cliente móvil            10 s
  balanceador capa 7        8 s
  servicio                  6 s
  conexión inactiva del servicio   75 s
  conexión inactiva del balanceador 60 s   ← menor, a propósito

  antes: el servicio cerraba a los 30 s y el balanceador
  mantenía 60 s → 502 intermitentes, ~180/día
  después: 502 intermitentes, 2/día
```

El resultado del año siguiente:

```
caídas totales por certificado                2          0
certificados sin renovación automática       19          0
certificados fuera de inventario              6          0
p99 del listado                           2.100 ms     235 ms
```

```
502 intermitentes al día                    180          2
destinos degradados retirados solos          no      35 s
peticiones cortadas por despliegue          ~400          0
```

La lección que esta clase deja: las dos caídas totales del año fueron por certificado, y ninguna por caducidad sorpresa: la renovación automática llevaba 71 días rota en un caso y no cubría un cuarto punto de entrada en el otro. La alerta que las habría evitado no es la de caducidad —que existía— sino la de antigüedad de la renovación. Y el problema de percentiles llevaba tres meses, costaba cero euros y se resolvió dejando de repartir por turnos.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-16-advanced-cloud-networking-edge/196-balanceo-l4-l7-proxies-tls-y-gestion-de-certificados/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 traffic-entry 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 traffic-entry 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 p99 es pésimo mientras el p50 está bien	Reparto rotatorio: una instancia degradada recibe la misma carga que las sanas	Reparte por menor número de conexiones en vuelo y añade expulsión de atípicos por tasa de error.
Todos los destinos aparecen como no sanos a la vez	La comprobación de disponibilidad es profunda y cayó una dependencia común	Separa /vivo de /listo y haz que /listo no dependa de dependencias blandas.
Errores 502 intermitentes sin causa aparente	El destino cierra las conexiones inactivas antes que el balanceador y este reutiliza una conexión muerta	Ordena los plazos cliente > balanceador > destino y haz que el destino mantenga la conexión inactiva más tiempo que el balanceador.
Cada despliegue corta peticiones en curso	No hay drenaje de conexiones o su plazo es menor que el de las peticiones largas	Configura drenaje con margen sobre el plazo más largo del servicio.
Un certificado caduca pese a tener renovación automática	La automatización llevaba semanas fallando en silencio	Alerta por antigüedad de la última renovación, no solo por caducidad, y prueba periódicamente que la alerta salta.
Solo falla una integración concreta con un tercero	Caducó un certificado de cliente que no está en ningún balanceador ni inventario	Inventaría todos los certificados, incluidos los de cliente y los de agentes, y ponlos bajo renovación automática.

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 puede hacer un balanceador de capa 7 que uno de capa 4 no?
2. ¿Por qué el reparto rotatorio empeora los percentiles?
3. ¿Qué diferencia hay entre el punto de vivo y el de listo?
4. ¿En qué orden deben estar los plazos y qué pasa si se invierten?
5. ¿Qué alerta detecta una renovación automática rota antes de que caduque el certificado?

Referencias

- AWS (2025). Elastic Load Balancing: types, health checks and connection draining. <https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html>
- Envoy (2025). Load balancing and outlier detection. <https://www.envoyproxy.io/docs/envoy/latest/intro/archoverview/upstream/loadbalancing/overview>
- RFC 8555 — Automatic Certificate Management Environment (ACME). <https://www.rfc-editor.org/rfc/rfc8555>
- Mozilla (2025). Server side TLS configuration guidelines. <https://wiki.mozilla.org/Security/ServerSideTLS>
- Google (2016). SRE Book: load balancing at the frontend. <https://sre.google/sre-book/load-balancing-frontend/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 16 en PDF · Manual integral

Anterior	Índice	Siguiente
← 195 · DNS autoritativo, recursivo, split-horizon y DNSSEC	Parte 16 · Programa	197 · CDN, caché, origin shielding y edge compute →

Evaluación — 196 Balanceo L4/L7, proxies, TLS y gestión de certificados

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 traffic-entry 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.

197 — CDN, caché, origin shielding y edge compute

Parte: 16 — Redes cloud avanzadas, conectividad híbrida y edge Nivel: avanzado · Horas estimadas: 4 Laboratorio: performance · Estado: EXECUTABLECORE

Propósito

Servir desde el borde lo que no hace falta calcular otra vez, que es la palanca más grande que existe sobre latencia y coste de salida a la vez. La clase explica qué determina de verdad la tasa de aciertos —la clave de caché, no el producto—, cómo se invalida sin tirar el origen, qué es y para qué sirve el escudo de origen, y dónde tiene sentido ejecutar código en el borde y dónde es una complicación cara.

Resultados de aprendizaje

Al finalizar podrás:

1. Diseñar la clave de caché para maximizar aciertos sin servir contenido incorrecto.
2. Elegir los tiempos de validez y la revalidación adecuados a cada contenido.
3. Invalidar sin provocar una avalancha contra el origen.
4. Aplicar el escudo de origen y medir lo que ahorra.
5. Decidir qué ejecutar en el borde y qué no.

Conceptos centrales

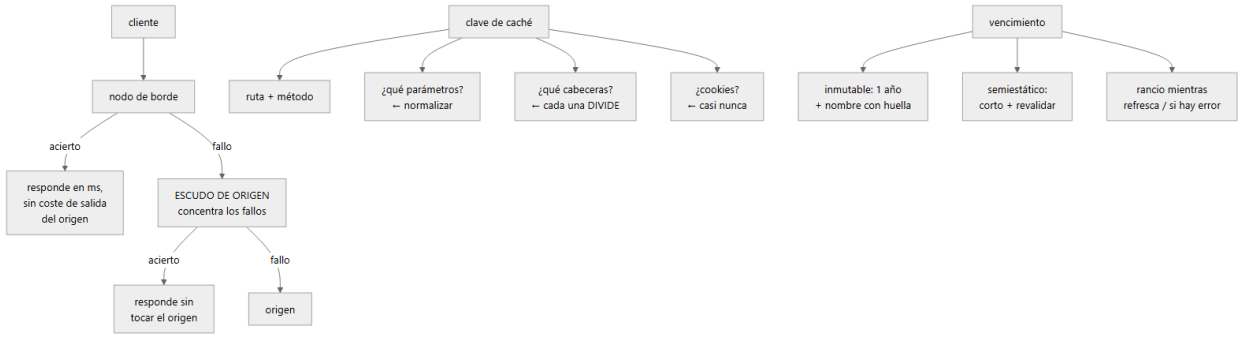
Concepto	Comprensión verificable
clave de caché	Conjunto de elementos de la petición que identifican una entrada. Determina la tasa de aciertos más que ningún otro factor.
tasa de aciertos	Proporción de peticiones servidas sin ir al origen. Se mide por ruta, no en global.
revalidación	Preguntar al origen si el contenido cambió sin volver a descargarlo, usando validadores.
contenido rancio	Servir una copia vencida mientras se refresca o cuando el origen falla. Convierte una caída en una degradación.
escudo de origen	Capa intermedia que concentra los fallos de caché de todos los nodos y protege al origen.
avalancha	Muchas peticiones simultáneas al origen tras una invalidación o un vencimiento común.

Modelo mental

La red cloud es un sistema distribuido de rutas, identidades y políticas; cada salto añade latencia, costo y una frontera de fallo.

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 clave de caché lo decide casi todo

La tasa de aciertos no depende del producto: depende de cuántas entradas distintas se generan para el mismo contenido.

```
CADA ELEMENTO QUE ENTRA EN LA CLAVE MULTIPLICA LAS ENTRADAS

  ruta                                  1 entrada
  + 3 valores de idioma                 3
  + 4 valores de moneda                12
  + cabecera de agente de usuario      ×miles   ← desastre
  + cookie de sesión                   ×usuarios ← desastre
```

Y los errores que hunden la tasa de aciertos, por frecuencia:

```
1. INCLUIR TODOS LOS PARÁMETROS DE CONSULTA
   ?id=7&utm_source=correo&fbclid=xyz
   → los de seguimiento crean una entrada por usuario
   solución   lista de parámetros que SÍ entran; el resto
              se ignora o se elimina

2. INCLUIR COOKIES
   una cookie de sesión o de analítica en la clave hace la
   caché inútil
   solución   no incluir cookies salvo en rutas concretas

3. INCLUIR EL AGENTE DE USUARIO
   miles de variantes
   solución   si hay que variar por dispositivo, usar una
              clasificación de 2-3 valores, no la cabecera

4. NO NORMALIZAR
   /producto?a=1&b=2 y /producto?b=2&a=1 son la misma cosa
   solución   ordenar y normalizar antes de construir la
              clave
```

Y el error simétrico, que es peor:

```
CLAVE DEMASIADO CORTA
  no incluir el idioma cuando el contenido varía por idioma
  → se sirve contenido incorrecto a alguien
  → y se sirve DESDE CACHÉ, así que dura hasta que vence

→ el mecanismo correcto es declarar la variación en la
  respuesta, para que la caché sepa qué diferencia
```

Y la medida que hay que mirar:

```
tasa de aciertos POR RUTA, no global
  la global la domina lo estático y esconde el problema
  → una ruta con 12 % de aciertos y mucho tráfico es donde
    está el dinero

y además: bytes servidos desde el borde frente a bytes desde
el origen
  → es lo que se traduce directamente en coste de salida
                                                  clase 168
```

2. Vencimiento, revalidación y contenido rancio

Hay tres decisiones distintas y suelen confundirse en un solo número.

```
1. CUÁNTO PUEDE SERVIRSE SIN PREGUNTAR      (validez)
2. QUÉ HACER CUANDO VENCE                   (revalidar)
3. QUÉ HACER SI EL ORIGEN NO RESPONDE       (rancio)
```

El patrón por tipo de contenido:

```
INMUTABLE (recursos con huella en el nombre)
  app.7f3a91.js
  validez   1 año, e inmutable
  cambio    nombre nuevo; no hay que invalidar nada
  → es la forma más eficaz y la más infrautilizada

SEMIESTÁTICO (páginas, listados, imágenes de producto)
  validez   corta (30-300 s)
  y además  rancio mientras refresca
  → el usuario nunca espera al origen

PERSONALIZADO
  no se cachea en el borde, o se cachea la parte común y se
  compone
  → separar la página en trozos cacheables y trozos propios

API DE LECTURA
  validez corta con revalidación por validador
  → respuesta 304 sin cuerpo: ahorra ancho de banda aunque
    no ahorre latencia

API DE ESCRITURA
  nunca
```

Contenido rancio, que es la función más valiosa y la menos configurada:

```
RANCIO MIENTRAS REFRESCA
  al vencer, se sirve la copia vieja INMEDIATAMENTE y se
  refresca por detrás
  → el usuario nunca paga la latencia del origen
  → y desaparece la avalancha por vencimiento simultáneo

RANCIO SI HAY ERROR
  si el origen devuelve error o no responde, se sirve la
  copia vieja
  → convierte una caída del origen en contenido con unos
    minutos de antigüedad
  → esto es lo que hace que la caída no se note      clase 185
```

Y el dato que justifica configurarlo:

```
un origen caído 20 minutos con rancio-si-error activo y
validez de 5 minutos
  → los usuarios ven contenido de hasta 25 minutos de
    antigüedad
  → en vez de una página de error
```

La avalancha, que es el fallo característico de las cachés:

```
CÓMO OCURRE
  todo se cachea a la vez (tras un despliegue o una
  invalidación) y vence a la vez
  → miles de peticiones simultáneas al origen
  → el origen cae, y al recuperarse vuelve a pasar

CÓMO SE EVITA
  rancio mientras refresca
  colapso de peticiones: una sola petición al origen por
    clave; las demás esperan esa
  dispersión del vencimiento: validez ± un porcentaje al azar
  y no invalidar todo a la vez              ← ver abajo
```

3. Invalidar y proteger el origen

La invalidación es la operación más peligrosa de una caché.

```
POR PREFIJO O COMODÍN
  /productos/*
  → invalida miles de entradas de golpe
  → todas las peticiones siguientes van al origen
  → es la causa de la avalancha de la clase 111

POR ETIQUETA
  cada respuesta lleva etiquetas: producto:7, categoría:3
  al cambiar el producto 7 se invalida solo lo suyo
  → es el mecanismo correcto y hay que diseñarlo desde el
    principio

POR NOMBRE NUEVO (versionado)
  no se invalida nada: el recurso nuevo tiene otro nombre
  → cero riesgo; es lo mejor cuando se puede
```

Y las reglas de operación:

```
una invalidación masiva se hace con el origen preparado
  → o escalonada, por lotes
  → nunca en hora punta
y se mide el efecto en la carga del origen
```

El escudo de origen, que resuelve un problema concreto:

```
SIN ESCUDO
  200 nodos de borde; cada uno falla su caché por separado
  → hasta 200 peticiones al origen por el mismo objeto

CON ESCUDO
  los fallos de todos los nodos van a una capa intermedia
  → el origen ve 1 petición por objeto

EFECTO TÍPICO
  reducción del tráfico al origen entre 3× y 10×
  y mejora de la tasa de aciertos global, porque el escudo
  acumula lo que cada nodo individualmente no vería
```

Y dónde ponerlo:

```
cerca del origen, no cerca de los usuarios
→ y si hay varias regiones de origen, un escudo por región
```

Lo que se ahorra, que es la justificación de todo el capítulo:

```
CADA ACIERTO EN EL BORDE
  no consume cómputo del origen
  no consume ancho de banda de salida del origen  ← el caro
  responde en 5-30 ms en vez de 80-300 ms

y el coste de salida desde el borde suele ser la mitad o
menos que desde el origen
→ subir 10 puntos la tasa de aciertos en la ruta de más
  tráfico suele valer más que cualquier optimización de
  código                                          clase 168
```

4. Cómputo en el borde: cuándo sí

Ejecutar código en los nodos de borde permite decidir cosas antes de llegar al origen. Es útil en un conjunto de casos concreto y una mala idea fuera de él.

```
DONDE SÍ COMPENSA
  reescribir rutas y normalizar la clave de caché
  redirigir por país, idioma o dispositivo
  autenticar o rechazar antes de llegar al origen
  componer una página con trozos cacheados
  pruebas A/B decidiendo la variante en el borde
  firmar o validar testigos de acceso a contenido
  añadir cabeceras de seguridad

DONDE NO
  lógica de negocio con estado
  acceso a base de datos
  procesos largos
  cualquier cosa que necesite consistencia fuerte
```

Y las limitaciones que hay que conocer antes de empezar:

```
tiempo de ejecución muy corto (milisegundos)
memoria pequeña
sin estado compartido fiable entre nodos
latencia alta hacia cualquier base de datos central
depuración difícil: cientos de nodos
despliegue global que tarda minutos en propagarse
```

Y la advertencia operativa:

```
un error en el código de borde afecta a CUANTO tráfico pasa a
la vez, y no hay despliegue escalonado natural
→ desplegar por porcentaje de tráfico si el producto lo
  permite, y tener vuelta atrás rápida    clase 102
```

Y una decisión de diseño que evita mucho trabajo:

```
lo que se pueda resolver con configuración de caché, no se
resuelve con código en el borde
→ normalizar parámetros, variar por idioma y añadir
  cabeceras suelen ser configuración
```

Y la lista de comprobación de la clase:

- la tasa de aciertos se mide por ruta, no en global
- los parámetros de seguimiento no entran en la clave
- no hay cookies en la clave salvo en rutas concretas
- los parámetros se normalizan y se ordenan
- la variación por idioma o dispositivo está declarada
- los recursos versionados usan huella en el nombre y validez larga
- está activado rancio mientras refresca
- está activado rancio si hay error
- hay colapso de peticiones por clave
- el vencimiento está disperso
- la invalidación es por etiqueta, no por comodín
- hay escudo de origen y se ha medido lo que ahorra
- se miden bytes servidos desde el borde y desde el origen
- el código de borde no hace lógica de negocio con estado

Y el cierre que enlaza con la clase siguiente: el borde resuelve el tráfico que viene de internet, pero buena parte del tráfico de una empresa va contra su propia red. Conectar la nube con los centros de datos —túneles, enlaces dedicados y su operación— es la materia de la clase 198.

Ejemplo trabajado

CloudShop tiene una tasa de aciertos global del 71 % y una factura de salida de 14.200 €/mes. Lo que sigue es el análisis por ruta, los cuatro cambios de clave de caché, y el resultado a los tres meses.

Lo que decía el panel:

```
tasa de aciertos global                          71 %
«está bien, la mayoría se sirve del borde»
```

Lo que decía el desglose por ruta:

```
ruta                    peticiones/día   aciertos   bytes al origen
/estáticos/*                12,4 M         99,4 %      1,2 %
/imagenes/producto/*         8,1 M         96,1 %      6,4 %
/api/catalogo/listado        4,7 M          8,2 %     41,3 %  ←
/producto/{id}               3,2 M         34,0 %     22,7 %  ←
/api/precios/{id}            2,9 M          0 %       11,8 %
/buscar                      1,1 M          0 %        9,3 %
resto                        0,8 M         62 %        7,3 %
```

Y la lectura:

```
el 71 % global lo produce lo estático, que ya estaba resuelto
las dos rutas que generan el 64 % del tráfico al origen
tienen tasas del 8 % y del 34 %
→ ahí está el dinero y la latencia
```

Diagnóstico de /api/catalogo/listado: 8,2 % de aciertos.

```
la clave incluía
```

```
  ruta
  TODOS los parámetros de consulta
  cabecera de agente de usuario
  cookie de sesión

lo que eso producía
  parámetros reales que cambian el contenido        3
    categoria, pagina, orden
  parámetros de seguimiento que llegaban           11
    utm_source, utm_medium, utm_campaign, fbclid, gclid…
  variantes de agente de usuario observadas    ~14.000
  cookies de sesión                          1 por usuario

→ entradas distintas para el mismo contenido: prácticamente
  una por petición
→ por eso el 8,2 %, que venía de recargas inmediatas
```

Los cuatro cambios.

```
CAMBIO 1 · lista blanca de parámetros
  entran   categoria, pagina, orden
  se ignoran los 11 de seguimiento y cualquiera nuevo
  además   se ordenan alfabéticamente antes de la clave

CAMBIO 2 · fuera el agente de usuario
  el contenido no variaba por dispositivo en esta ruta
  → eliminado de la clave

CAMBIO 3 · fuera las cookies
  el listado no es personalizado
  la personalización (los 3 productos recomendados) se
  extrajo a una llamada aparte que no se cachea
  → la página común se cachea; la parte propia, no

CAMBIO 4 · validez y rancio
  validez              60 s
  rancio mientras refresca   600 s
  rancio si hay error      86.400 s (1 día)
  dispersión del vencimiento  ±15 %
  colapso de peticiones    activado
```

Y el resultado de esa ruta:

```
tasa de aciertos                      8,2 %       94,1 %
p50 de la ruta                       112 ms         9 ms
p99 de la ruta                       890 ms        38 ms
peticiones/día al origen             4,31 M       0,28 M
```

Diagnóstico de /producto/{id}: 34 % de aciertos.

```
causa    invalidación por comodín
         cada cambio de precio invalidaba /producto/*
         y los precios cambian 11 veces al mes… por producto
         → en la práctica, invalidaciones masivas casi diarias

y el efecto secundario
  tras cada invalidación, avalancha contra el origen
  → 3 incidentes de latencia en el año coincidían con
    invalidaciones                                clase 111

corrección
  etiquetado de respuestas: producto:{id}, categoria:{id}
  invalidación por etiqueta: cambiar el producto 4471
  invalida solo lo suyo
  y la invalidación por comodín se desactivó por política

resultado                             antes      después
tasa de aciertos                       34 %        88 %
entradas invalidadas por cambio     ~310.000          4
avalanchas contra el origen           3/año           0
```

El escudo de origen.

```
antes   214 nodos de borde, cada uno fallando por su cuenta
        peticiones al origen por objeto nuevo: hasta 190
```

```
después escudo en la región del origen
        peticiones al origen por objeto nuevo: 1

efecto medido
  peticiones/s al origen en el pico     3.100  →  410
  ancho de banda de salida del origen   -78 %
  y un efecto no previsto: la tasa de aciertos global subió
    2,3 puntos más, porque el escudo acumula lo que un nodo
    con poco tráfico nunca llegaría a cachear
```

Las dos rutas que no se cachearon, y por qué:

```
/api/precios/{id}
  se evaluó y se descartó cachear en el borde
  motivo   el precio ya se cachea 15 min en la aplicación
           (clase 187) y el volumen al origen es pequeño
  además   una caché más añade otra capa donde el precio
           puede quedarse viejo, y ya hay una decisión de
           consistencia tomada

/buscar
  las consultas son casi únicas: la tasa de aciertos
  estimada era del 4 %
  → no compensa
```

Y esa segunda decisión merece nota:

```
se midió antes de configurar
→ configurar caché en una ruta con 4 % de aciertos añade
  complejidad y no ahorra nada
```

El cómputo en el borde, donde se usó y donde no:

```
SÍ
  normalización de parámetros y de mayúsculas en la ruta
  redirección por país a la tienda correspondiente
  validación del testigo firmado para descargas privadas
    → rechaza el 100 % de las peticiones sin testigo válido
      sin tocar el origen
  cabeceras de seguridad

NO
  se propuso mover el cálculo de recomendaciones al borde
  → necesitaba consultar un almacén central en cada petición
  → la latencia hacia el almacén desde un nodo lejano era
    de 90-140 ms, peor que ir al origen
  → descartado, con el registro correspondiente  clase 190
```

El resultado global, a los tres meses:

```
tasa de aciertos global                     71 %       97,2 %
tasa de aciertos de /api/catalogo/listado  8,2 %       94,1 %
tasa de aciertos de /producto/{id}          34 %         88 %
peticiones/s al origen en el pico          3.100         410
p50 del listado en el borde               112 ms        9 ms
coste de salida                         14.200 €/mes  4.900 €/mes
coste de cómputo del origen              -3.400 €/mes (2 zonas
                                          menos de instancias)
avalanchas                                   3/año          0
```

Y el detalle que resume el capítulo:

```
del ahorro de 9.300 €/mes, la mayor parte vino de un cambio
que no tocó ni una línea de código de la aplicación:
dejar de incluir en la clave once parámetros de seguimiento,
el agente de usuario y una cookie
```

La lección que esta clase deja: la tasa de aciertos global del 71 % escondía dos rutas al 8 % y al 34 % que generaban el 64 % del tráfico al origen, y ninguna de las dos tenía un problema de producto: una tenía la clave de caché mal construida y la otra se invalidaba con comodín. Y el mayor ahorro del año se consiguió quitando cosas de la clave, no añadiendo capacidad.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-16-advanced-cloud-networking-edge/197-cdn-cache-origin-shielding-y-edge-compute/lab.py
```

El laboratorio selecciona el motor de práctica performance 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 cdn-experiment 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 prueba de carga con baseline y cuello de botella. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye cdn-experiment 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 tasa de aciertos es alta en global y el origen recibe mucho tráfico	La global la domina el contenido estático y esconde rutas con tasas bajas	Mide aciertos y bytes al origen por ruta, y ataca las rutas con más tráfico y peor tasa.
Casi cada petición genera una entrada nueva	La clave incluye parámetros de seguimiento, cookies o el agente de usuario	Usa lista blanca de parámetros, normaliza y ordena, y saca cookies y agente de usuario de la clave.
Se sirve contenido incorrecto a algunos usuarios	La clave es demasiado corta y no distingue una variación real	Declara la variación en la respuesta para que la caché sepa qué diferencia, en vez de ampliar la clave a ciegas.
Tras cada invalidación el origen se satura	Invalidación por comodín y vencimiento simultáneo	Invalida por etiqueta, activa rancio mientras refresca, colapso de peticiones y dispersión del vencimiento.
Una caída del origen se ve como página de error	No está activado servir contenido rancio si hay error	Configura rancio si hay error con una ventana amplia; convierte la caída en contenido con algo de antigüedad.
El código en el borde resulta más lento que ir al origen	Necesita consultar un almacén central en cada petición	Reserva el borde para decisiones sin estado; lo que dependa de datos centrales, déjalo en el origen.

■ 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 factor determina la tasa de aciertos más que ningún otro?
2. ¿Cómo se sirve contenido que varía por idioma sin destrozar la caché?
3. ¿Qué diferencia hay entre rancio mientras refresca y rancio si hay error?
4. ¿Por qué la invalidación por comodín provoca avalanchas y qué la sustituye?
5. ¿Qué problema concreto resuelve el escudo de origen?

Referencias

• RFC 9111 — HTTP caching. <https://www.rfc-editor.org/rfc/rfc9111>

• RFC 5861 — HTTP cache-control extensions for stale content. <https://www.rfc-editor.org/rfc/rfc5861>

• AWS (2025). CloudFront: cache key and origin shield. <https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/understanding-the-cache-key.html>

• Fastly (2025). Purging with surrogate keys. <https://docs.fastly.com/en/guides/working-with-surrogate-keys>

• Cloudflare (2025). Cloudflare Workers: what edge compute is good for. <https://developers.cloudflare.com/workers/>

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

Descargar: Parte 16 en PDF · Manual integral

Anterior	Índice	Siguiente
← 196 · Balanceo L4/L7, proxies, TLS y gestión de certificados	Parte 16 · Programa	198 · VPN, Direct Connect, ExpressRoute e Interconnect →

Evaluación — 197 CDN, caché, origin shielding y edge compute

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 cdn-experiment 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.

198 — VPN, Direct Connect, ExpressRoute e Interconnect

Parte: 16 — Redes cloud avanzadas, conectividad híbrida y edge Nivel: avanzado · Horas estimadas: 4 Laboratorio: network · Estado: EXECUTABLECORE

Propósito

Conectar la nube con los centros de datos y las oficinas eligiendo bien entre túnel por internet y enlace dedicado, que se decide casi siempre por costumbre y casi nunca por cifras. La clase compara las dos opciones por lo que de verdad las diferencia —previsibilidad, plazo de provisión y coste por gigabyte—, explica cómo se construye la redundancia que sí protege, y detalla los fallos de operación que producen la mayoría de los incidentes: túneles con parámetros desalineados, ancho de banda mal dimensionado y enlaces sin conmutación probada.

Resultados de aprendizaje

Al finalizar podrás:

1. Elegir entre túnel e enlace dedicado con criterios medibles.
2. Dimensionar el ancho de banda contando lo que de verdad va a cruzar.
3. Diseñar redundancia sin causas comunes ocultas.
4. Operar túneles y enlaces con la vigilancia que evita los fallos típicos.
5. Probar la conmutación antes de necesitarla.

Conceptos centrales

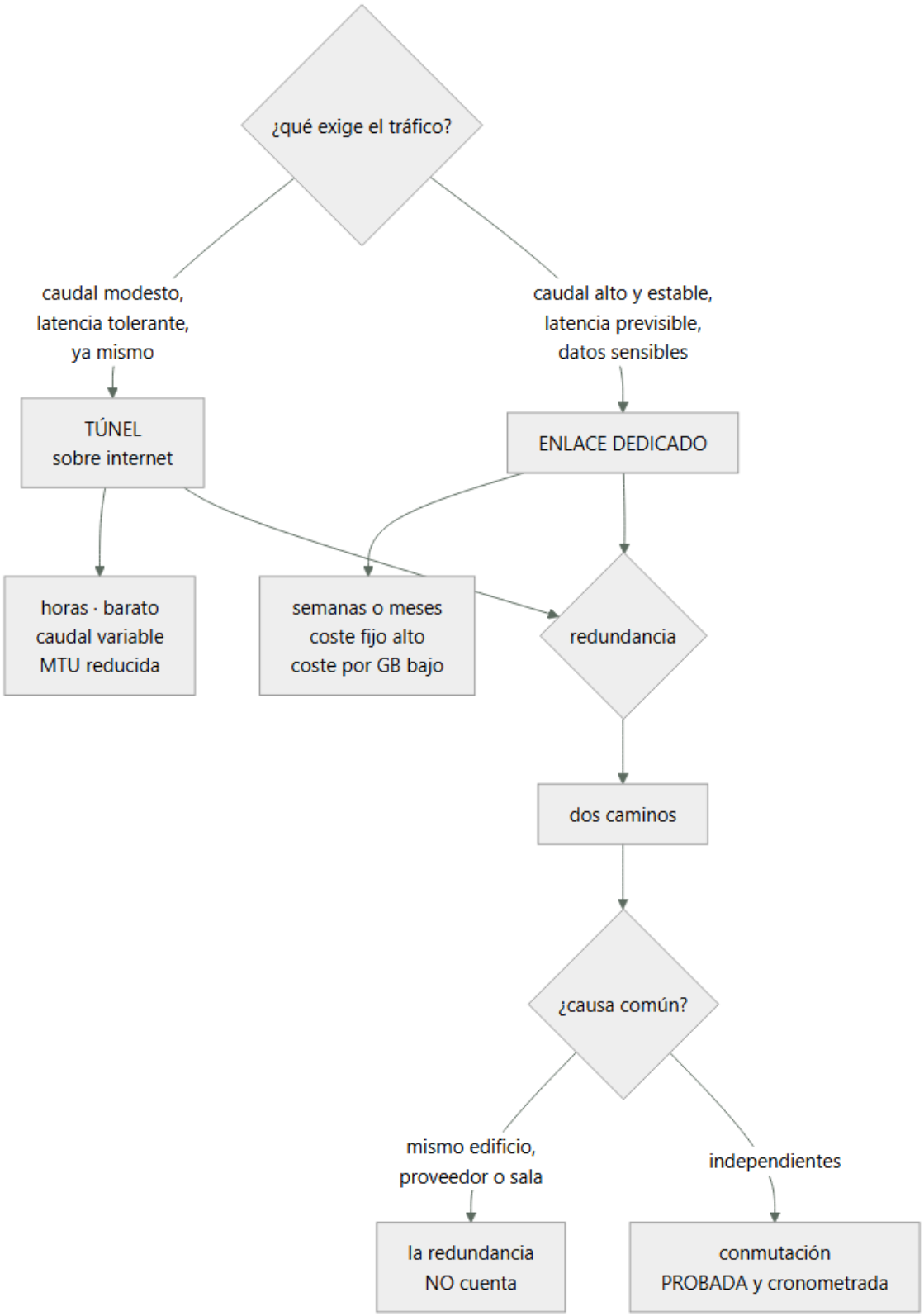
Concepto	Comprensión verificable
túnel sobre internet	Conexión cifrada entre dos redes a través de internet. Barata, rápida de montar, con latencia y caudal variables.
enlace dedicado	Circuito físico privado hacia el proveedor. Previsible y con caudal garantizado; caro y lento de provisionar.
asociación virtual	Circuito lógico sobre un enlace físico, que separa el tráfico privado del de servicios públicos.
MTU y fragmentación	Tamaño máximo de paquete. Un túnel lo reduce, y si no se ajusta produce fallos parciales difíciles de diagnosticar.
causa común	Elemento compartido por dos caminos supuestamente redundantes: mismo edificio, mismo proveedor, misma sala.
conmutación probada	Cambio al camino de respaldo ejecutado de verdad y cronometrado, no descrito en un documento.

Modelo mental

La red cloud es un sistema distribuido de rutas, identidades y políticas; cada salto añade latencia, costo y una frontera de fallo.

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. Túnel o enlace dedicado

Las dos opciones resuelven lo mismo y se diferencian en cinco cosas que se pueden medir.

```
                    TÚNEL             ENLACE DEDICADO
plazo de provisión  horas             4-12 semanas
coste fijo          bajo (~50-300 €)  alto (500-5.000 €/mes
```

```
                                    + puerto + cruce)
coste por GB          el de salida  claramente menor
                      estándar      (a menudo la mitad
                                    o menos)
caudal                limitado por la garantizado por
                      pasarela      contrato
                      (1-10 Gbps según
                      producto)
latencia              variable,     estable,
                      depende de    previsible
                      internet
jitter                alto          bajo
```

Y el criterio que decide de verdad:

```
¿el tráfico es SENSIBLE A LA VARIABILIDAD?
  replicación de bases, escritorios remotos, voz, sistemas
  transaccionales antiguos
  → enlace dedicado

¿el volumen mensual es alto?
  calcula el punto de equilibrio:
    coste fijo del enlace / (ahorro por GB)
  → a partir de cierto volumen, el enlace sale más barato
    aunque no se necesite su previsibilidad

¿hace falta ya y el volumen es modesto?
  → túnel, y se decide después con datos
```

Y el patrón que evita elegir mal:

```
empieza con túnel, mide durante 2-3 meses, y decide
→ y deja el túnel como respaldo del enlace cuando llegue
→ así el enlace no nace sin redundancia
```

Las asociaciones virtuales, que confunden al principio:

```
un enlace físico transporta VARIOS circuitos lógicos
  uno privado hacia tu red virtual
  uno público hacia los servicios del proveedor sin pasar
    por internet                                  clase 200
  uno de tránsito hacia una pasarela central      clase 199

→ y cada uno tiene su propia sesión BGP y sus rutas
                                                  clase 194
```

Y una decisión que ahorra dinero y casi nadie hace:

```
el tráfico hacia los almacenes de objetos y otros servicios
del proveedor puede ir por el circuito público del enlace
→ evita el coste de salida a internet
→ pero ojo: el circuito público anuncia MUCHOS prefijos
  → y ahí es donde se supera el límite de la sesión BGP
                                                  clase 194
```

2. Dimensionar y la trampa de la MTU

El dimensionado se hace mal por contar solo lo que se ve.

```
LO QUE CRUZA Y NADIE CUENTA
  copias de seguridad y su ventana nocturna
  replicación de bases de datos                   clase 161
  sincronización de ficheros y de imágenes
  telemetría y registros hacia el recolector central
  actualizaciones de sistema operativo
  escritorios remotos, si los hay
  y el pico de una migración, que puede ser 20× lo normal

REGLA
  dimensiona por el percentil 95 del tráfico, no por la media
  → los proveedores facturan así, y el codo también está ahí
                                                  clase 186
```

Y el error de saturación, que se manifiesta raro:

```
un enlace saturado no «va más lento» de forma uniforme
  se llenan las colas de los equipos
```

```
  sube la latencia y el jitter
  se descartan paquetes
  → y lo primero que falla es lo sensible a la variabilidad,
    no lo que consume el ancho de banda

→ síntoma típico: la replicación va bien y las sesiones
  interactivas se cortan
```

La MTU, que es la causa de una familia entera de fallos difíciles:

```
un túnel AÑADE cabeceras y por tanto REDUCE el tamaño útil
  típico   1.500 → 1.436 o menos según el tipo de túnel

SI NO SE AJUSTA
  los paquetes grandes no caben y hay que fragmentar o
  descartar
  el mecanismo estándar de descubrimiento depende de mensajes
  de control que muchos cortafuegos bloquean
  → y entonces los paquetes se descartan EN SILENCIO

LA FIRMA DEL FALLO
  las conexiones se establecen (paquetes pequeños)
  las peticiones pequeñas funcionan
  las respuestas grandes se cuelgan
  el fichero de 2 KB baja y el de 2 MB no
  y a veces funciona con un cliente y no con otro
```

Y las correcciones, por orden:

```
1  permitir los mensajes de control necesarios en los
   cortafuegos    ← lo correcto
2  ajustar la MTU en las interfaces de los extremos
3  ajustar el tamaño máximo de segmento en el punto de
   entrada del túnel
→ y comprobarlo con un paquete grande sin fragmentar, no
  suponerlo
```

3. Redundancia que sí protege

Duplicar el enlace es fácil; conseguir que los dos no caigan a la vez, no.

```
LAS CAUSAS COMUNES QUE ANULAN LA REDUNDANCIA  clase 185
  los dos circuitos entran por el mismo edificio
  los dos usan el mismo proveedor de última milla
  los dos terminan en la misma sala o en el mismo equipo
  los dos van por la misma zanja      ← ocurre más de lo que
                                        parece
  los dos usan el mismo router de la nube
  las dos sesiones BGP tienen el mismo límite de prefijos
    → se caen juntas por el mismo motivo            clase 194
```

Y las preguntas que hay que hacer al proveedor, por escrito:

```
¿por qué ruta física va cada circuito?
¿comparten alguna sección? ¿cuál?
¿en qué sala termina cada uno?
¿qué proveedor de última milla usa cada uno?
→ y si no contestan con precisión, la redundancia es una
  suposición
```

Los niveles de redundancia, con lo que cubre cada uno:

```
UN ENLACE, SIN NADA
  → un corte es una caída total; plazo de reparación de
    horas o días

UN ENLACE + TÚNEL DE RESPALDO
  cubre el corte del enlace, con caudal y latencia peores
  → suficiente para la mayoría, y es lo más rentable
  → exige probar la conmutación                    ley 22

DOS ENLACES, MISMA UBICACIÓN
  cubre el fallo de un puerto o un equipo
  no cubre el corte de la fibra ni el fallo del edificio

DOS ENLACES, UBICACIONES DISTINTAS
```

```
  cubre casi todo
  → y es lo que exigen los acuerdos de nivel de servicio
    más altos de los proveedores

DOS ENLACES + DOS REGIONES DE NUBE
  para cuando el objetivo de continuidad lo exige
                                                clase 166
```

Y una advertencia sobre los acuerdos de nivel de servicio:

```
el proveedor promete un porcentaje SOLO si cumples su
topología de referencia
→ un solo enlace no tiene compromiso, aunque se pague
→ leerlo antes de prometer disponibilidad aguas arriba
                                                clase 185
```

Y el reparto del tráfico entre dos caminos:

```
ACTIVO-PASIVO   más simple; el respaldo se prueba poco
                → y por eso falla cuando hace falta
ACTIVO-ACTIVO   ambos en uso; el fallo se nota antes
                → pero exige cuidar la simetría     clase 194

y si es activo-pasivo, ejercitar el pasivo periódicamente
con tráfico real es lo único que garantiza que funcione
```

4. Operación y prueba

Los enlaces y túneles fallan poco, y por eso su operación se descuida.

```
LO QUE HAY QUE VIGILAR
  estado del circuito físico (luz, errores de trama)
  estado de la sesión BGP y prefijos recibidos      clase 194
  utilización, en percentil 95, contra el contratado
  latencia y jitter entre extremos, continuamente
  pérdida de paquetes
  y para túneles: estado de la asociación de seguridad y
    renegociaciones
```

Y las alertas que más incidentes evitan:

```
«utilización por encima del 70 % sostenida»
  → antes de que se note en la latencia
«el respaldo lleva N días sin llevar tráfico»       ley 13
  → detecta el respaldo que ya no funciona
«la latencia ha cambiado de forma sostenida»
  → suele significar que se conmutó a otro camino sin
    que nadie lo supiera
```

Y esa última merece detalle:

```
un cambio de latencia de 4 ms a 28 ms sin aviso significa
que el tráfico está pasando por otro sitio
→ y muy a menudo, por el respaldo, sin que nadie lo sepa
→ el enlace principal lleva días caído y no ha saltado
  ninguna alerta                                    ley 15
```

Los parámetros de túnel desalineados, que causan cortes periódicos:

```
SÍNTOMA CLÁSICO
  el túnel se cae y se levanta cada 8 horas, o cada 24
  → tiempos de vida de las claves distintos en cada extremo
  → o algoritmos negociados de forma distinta

OTROS PARÁMETROS QUE HAY QUE ALINEAR
  tiempos de vida de fase 1 y fase 2
  detección de par muerto y sus umbrales
  quién inicia la renegociación
  selectores de tráfico

→ y la configuración de los dos extremos debe estar en el
  mismo repositorio, no en dos consolas distintas
                                                clase 190
```

Las pruebas negativas de esta clase:

- desconectar el enlace principal y cronometrar la

conmutación
- comprobar que tras conmutar, la ida y la vuelta siguen siendo simétricas clase 194
- enviar un paquete grande sin fragmentar por el túnel
- saturar el enlace al 100 % y ver qué se degrada primero
- dejar el respaldo llevando tráfico real durante 24 h
- simular la caída de una sesión BGP y medir el restablecimiento
- volver al principal y comprobar que no queda tráfico asimétrico

Y la lista de comprobación de la clase:

- la elección entre túnel y enlace está justificada con cifras de volumen y de variabilidad
- el dimensionado usa el percentil 95 e incluye copias, replicación y telemetría
- la MTU está ajustada y comprobada con paquete grande
- está documentada la ruta física de cada circuito
- no hay causa común entre los dos caminos
- el acuerdo de nivel de servicio corresponde a la topología desplegada
- hay alerta de utilización al 70 %
- hay alerta de respaldo sin tráfico durante N días
- hay alerta de cambio sostenido de latencia
- los parámetros de túnel están alineados y versionados
- la conmutación se ha ejecutado y cronometrado

Y el cierre que enlaza con la clase siguiente: con varias redes, varias cuentas y varios enlaces, conectar cada cosa con cada cosa deja de funcionar. Las pasarelas de tránsito y los concentradores de conectividad son la materia de la clase 199.

Ejemplo trabajado

CloudShop conecta dos centros de datos y once oficinas con tres nubes. Lo que sigue es la decisión entre túnel y enlace con sus cifras, el incidente de MTU que duró cinco meses, y la redundancia que resultó no serlo.

La decisión, con datos de tres meses de túnel.

```
tráfico medido entre el centro de datos principal y la nube

  medio                                  340 Mbps
  percentil 95                           820 Mbps
  pico durante la ventana de copias     1,9 Gbps
  volumen mensual                         94 TB

  desglose del percentil 95
    replicación de la base heredada        41 %
    copias de seguridad nocturnas          27 %
    telemetría hacia el recolector         14 %
    tráfico de aplicación                  11 %
    actualizaciones y otros                 7 %

y la variabilidad observada con el túnel
  latencia p50                              9 ms
  latencia p99                             84 ms
  jitter                                hasta 60 ms
  pérdida de paquetes                     0,3 %
```

Y las dos consecuencias que decidieron:

```
1  la replicación de la base heredada se retrasaba de forma
   irregular; el retraso llegaba a 11 minutos en horas de
   internet congestionado
   → el objetivo de pérdida era de 1 minuto        clase 166
   → el túnel NO podía cumplirlo

2  el coste
   94 TB/mes por salida a internet         ~7.500 €/mes
   enlace dedicado de 1 Gbps
     puerto + cruce + circuito              2.900 €/mes
     coste por GB del enlace                ~1.900 €/mes
     total                                   4.800 €/mes
```

```
   → ahorro de 2.700 €/mes ADEMÁS de la previsibilidad

decisión   enlace dedicado de 1 Gbps, con el túnel
           existente como respaldo
plazo      9 semanas de provisión
```

Y una decisión adicional que salió del desglose:

```
las copias de seguridad no necesitaban ir al centro de datos
→ se reconfiguraron para escribir directamente en el
  almacén de objetos de la nube por el circuito público
  del enlace
→ el percentil 95 bajó de 820 a 520 Mbps
→ y el enlace de 1 Gbps pasó de justo a holgado
```

El incidente de MTU, que llevaba cinco meses.

```
síntoma reportado desde marzo
  «la exportación de informes desde la oficina de Lisboa
   falla a veces»
  se atribuyó a la aplicación; se revisó tres veces sin
  encontrar nada

lo que de verdad ocurría
  las peticiones pequeñas funcionaban siempre
  las respuestas de más de ~1,4 KB se colgaban
  el navegador esperaba hasta el plazo y fallaba

diagnóstico, en agosto
  el túnel de Lisboa tenía MTU efectiva de 1.398
  las interfaces estaban en 1.500
  el descubrimiento automático no funcionaba porque el
  cortafuegos de la oficina bloqueaba los mensajes de
  control necesarios
  → los paquetes grandes se descartaban EN SILENCIO

por qué tardó cinco meses
  el síntoma parecía de aplicación
  funcionaba desde las otras diez oficinas
  y funcionaba con ficheros pequeños, así que las pruebas
  manuales pasaban

corrección
  permitir los mensajes de control en el cortafuegos
  ajustar el tamaño máximo de segmento en el punto de
  entrada del túnel
  comprobación negativa nueva: enviar un paquete de 1.472
  bytes sin fragmentar desde CADA oficina, semanalmente
  → 2 oficinas más fallaron la primera ejecución
```

La redundancia que no lo era.

```
en noviembre se contrataron DOS enlaces de 1 Gbps «para
tener redundancia»
  proveedor A, circuito 1
  proveedor A, circuito 2

la prueba negativa de diciembre
  se pidió al proveedor la ruta física de cada circuito
  respuesta, tras insistir dos semanas
    los dos circuitos entran por el MISMO edificio
    comparten los primeros 2,1 km de conducción
    y terminan en la MISMA sala del punto de presencia

→ una excavadora en esa calle habría cortado los dos
→ la redundancia no cubría el escenario más probable

corrección
  el segundo circuito se movió a otro proveedor de última
  milla y a otro punto de presencia
  coste adicional                          +610 €/mes
  y se documentó la ruta física de ambos, por escrito
```

Y el hallazgo adicional:

```
las dos sesiones BGP tenían el mismo límite de prefijos
→ si la corporativa volvía a anunciar de más, se caían las
  dos a la vez, como en el incidente de marzo   clase 194
→ es una causa común que no es física
→ corregido con filtros y resumen en el origen
```

La conmutación, probada.

```
primera ejecución, enero
  se desconectó el enlace principal a las 22:00

  detección de la caída                        14 s
  reconvergencia BGP                           41 s
  tráfico restablecido por el respaldo        55 s   ✓
  PERO
    la replicación de la base se detuvo y no se reanudó
    → su cliente tenía un plazo de 30 s y no reintentaba
    → hubo que reiniciarla a mano
    → 6 minutos de retraso acumulado

  y al VOLVER al principal
    durante 4 minutos, la ida iba por el principal y la
    vuelta por el respaldo
    → el cortafuegos con estado descartó tráfico
    → asimetría, exactamente como en la clase 194

correcciones
  reintento con retroceso en el cliente de replicación
  preferencias alineadas en ambos extremos para la vuelta
  procedimiento de vuelta escrito y probado por otra
  persona                                          ley 22

segunda ejecución, abril
  conmutación                                  52 s
  replicación                          reanudada sola
  vuelta al principal                  sin asimetría
  todo correcto
```

La vigilancia que se montó después:

```
utilización p95 contra contratado, alerta al 70 %
latencia y jitter entre extremos, continuos
alerta por cambio sostenido de latencia
  → en marzo detectó que el tráfico llevaba 3 días por el
    respaldo sin que nadie lo supiera: el principal estaba
    caído desde una intervención del proveedor    ley 15
alerta «el respaldo lleva más de 45 días sin tráfico»
  → obliga a ejercitarlo
estado de sesiones BGP y prefijos recibidos
prueba de paquete grande sin fragmentar, semanal, por oficina
```

El resultado del año siguiente:

```
retraso máximo de replicación             11 min       38 s
incidentes atribuidos a «la aplicación»
  que eran de red                             3          0
coste mensual de conectividad          7.500 €     5.410 €
tiempo de conmutación                   sin medir      52 s
días con tráfico por el respaldo sin
  que nadie lo supiera                        3          0
circuitos con ruta física documentada         0          2
```

La lección que esta clase deja: la decisión entre túnel y enlace se tomó con tres meses de datos y resultó ahorrar dinero además de resolver el problema de variabilidad, que era lo que se buscaba. El incidente que más tiempo consumió —cinco meses— no parecía de red: era una MTU sin ajustar cuyo síntoma era «la exportación falla a veces». Y la redundancia contratada en noviembre no era redundancia: dos circuitos del mismo proveedor por la misma zanja, y hacía falta preguntarlo por escrito para saberlo.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-16-advanced-cloud-networking-edge/198-vpn-direct-connect-expressroute-e-
```

```
interconnect/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 hybrid-connectivity 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 hybrid-connectivity 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 replicación se retrasa de forma irregular pese a tener ancho de banda de sobra	El tráfico va por un túnel sobre internet, cuya latencia y jitter varían	Mide latencia, jitter y pérdida durante meses; si el tráfico es sensible a la variabilidad, pasa a enlace dedicado.
Las peticiones pequeñas funcionan y las respuestas grandes se cuelgan	MTU reducida por el túnel y descubrimiento automático bloqueado por un cortafuegos	Permite los mensajes de control, ajusta MTU y tamaño máximo de segmento, y comprueba con un paquete grande sin fragmentar desde cada sede.
El enlace saturado corta las sesiones interactivas mientras la replicación va bien	La saturación degrada primero lo sensible a latencia y jitter, no lo que consume el ancho de banda	Dimensiona por el percentil 95 contando copias, replicación y telemetría, y alerta al 70 % de utilización.
Los dos circuitos redundantes caen a la vez	Causa común: mismo edificio, misma zanja, mismo proveedor de última milla o mismo límite de prefijos	Exige por escrito la ruta física de cada circuito y separa proveedor, edificio y punto de presencia; revisa también las causas comunes lógicas.
El tráfico lleva días pasando por el respaldo y nadie lo sabe	No hay alerta de cambio sostenido de latencia ni de estado del camino principal	Alerta por variación sostenida de latencia y por respaldo sin tráfico durante N días.
El túnel se cae y se levanta con periodicidad exacta	Tiempos de vida o algoritmos desalineados entre los dos extremos	Alinea los parámetros de ambas fases y guarda la configuración de los dos extremos en el mismo repositorio.

■ Seguridad, ética y costo

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

■ Preguntas de comprobación

1. ¿Qué cinco factores diferencian un túnel de un enlace dedicado?
2. ¿Por qué el dimensionado se hace por percentil 95 y qué se olvida contar?
3. ¿Cuál es la firma característica de un problema de MTU?
4. ¿Qué preguntas hay que hacer al proveedor para saber si la redundancia es real?
5. ¿Qué alerta detecta que el tráfico lleva días por el camino de respaldo?

Referencias

- AWS (2025). Direct Connect resiliency recommendations. <https://docs.aws.amazon.com/directconnect/latest/UserGuide/highresiliency.html>
- Microsoft (2025). ExpressRoute design for high availability. <https://learn.microsoft.com/en-us/azure/expressroute/designing-for-high-availability-with-expressroute>
- Google Cloud (2025). Cloud Interconnect and HA VPN topologies. <https://cloud.google.com/network-connectivity/docs/interconnect>
- RFC 4459 — MTU and fragmentation issues with in-the-network tunneling. <https://www.rfc-editor.org/rfc/rfc4459>
- RFC 7296 — Internet Key Exchange Protocol Version 2 (IKEv2). <https://www.rfc-editor.org/rfc/rfc7296>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 16 en PDF · Manual integral

Anterior	Índice	Siguiente
← 197 · CDN, caché, origin shielding y edge compute	Parte 16 · Programa	199 · Transit Gateway, Virtual WAN y Network Connectivity Center →

Evaluación — 198 VPN, Direct Connect, ExpressRoute e Interconnect

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 hybrid-connectivity 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.

199 — Transit Gateway, Virtual WAN y Network Connectivity Center

Parte: 16 — Redes cloud avanzadas, conectividad híbrida y edge Nivel: avanzado · Horas estimadas: 4 Laboratorio: network · Estado: EXECUTABLECORE

Propósito

Sustituir la maraña de conexiones punto a punto por un concentrador de conectividad, y hacerlo sin convertirlo en un punto único ni en una autopista donde todo alcanza a todo. La clase explica por qué el emparejamiento directo deja de escalar, cómo se usan las tablas de rutas del concentrador para segmentar de verdad, qué cuesta cada salto en dinero y latencia, y cuándo un concentrador es la respuesta equivocada.

Resultados de aprendizaje

Al finalizar podrás:

1. Explicar por qué las conexiones punto a punto no escalan.
2. Segmentar con tablas de rutas del concentrador en vez de con reglas.
3. Calcular el coste de un concentrador en dinero y en latencia.
4. Conectar varias regiones y varias nubes sin crear caminos indeseados.
5. Decidir cuándo NO usar un concentrador.

Conceptos centrales

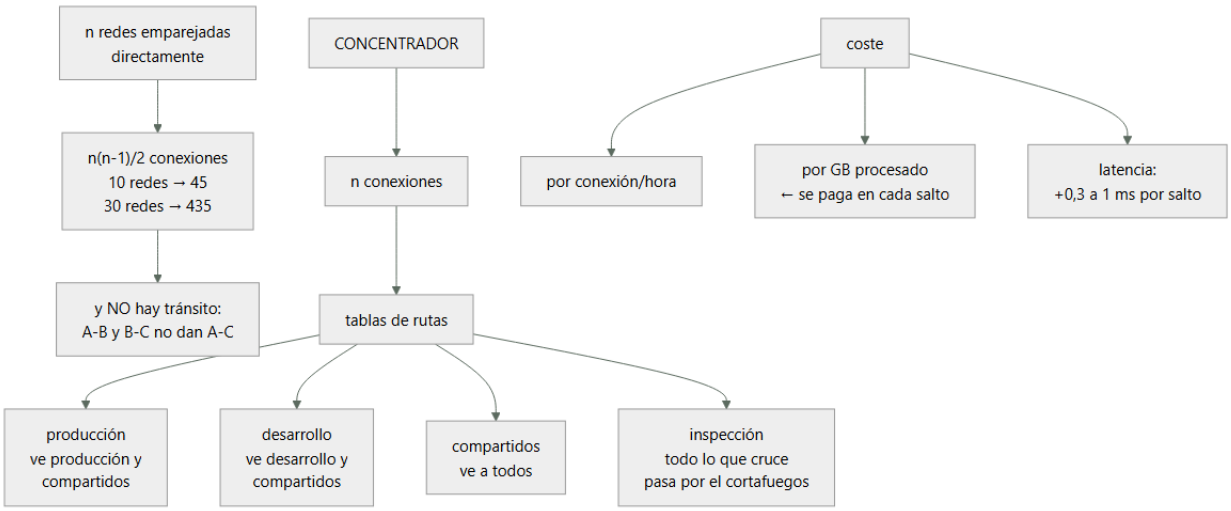
Concepto	Comprensión verificable
emparejamiento directo	Conexión entre dos redes. Simple, barata y sin tránsito: no encadena.
concentrador de conectividad	Servicio central al que se conectan las redes y que encamina entre ellas.
tabla de rutas del concentrador	Conjunto de rutas asociado a un grupo de conexiones. Es el mecanismo de segmentación real.
aislamiento por asociación	Que una red solo vea las rutas de la tabla a la que está asociada.
inspección centralizada	Hacer pasar el tráfico entre segmentos por un cortafuegos común.
coste por salto	Lo que cobra el concentrador por conexión y por gigabyte procesado, que se paga en cada dirección.

Modelo mental

La red cloud es un sistema distribuido de rutas, identidades y políticas; cada salto añade latencia, costo y una frontera de fallo.

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

■ Flujo de razonamiento



Desarrollo

1. Por qué el emparejamiento directo se acaba

Conectar dos redes directamente es lo más simple y lo más barato, y funciona muy bien hasta que deja de hacerlo.

```
LO BUENO DEL EMPAREJAMIENTO DIRECTO
  sin salto intermedio: latencia mínima
  sin coste de procesamiento por gigabyte
  sin punto único añadido

LO QUE LO ROMPE
  1  NO HAY TRÁNSITO
     si A-B y B-C están emparejadas, A NO alcanza C
     → hay que emparejar A-C explícitamente

  2  CRECIMIENTO CUADRÁTICO
     n(n-1)/2 conexiones
       5 redes  →  10
      10 redes  →  45
      30 redes  → 435

  3  RUTAS EN CADA TABLA
     cada red necesita una ruta por cada otra red
     → se llega al límite de rutas por tabla       clase 194

  4  IMPOSIBLE DE CAMBIAR
     añadir una red exige tocar n tablas
```

Y el umbral práctico:

```
hasta ~6-8 redes, emparejar directamente sigue siendo la
opción más simple y más barata
por encima, un concentrador se paga solo en trabajo evitado

y el patrón mixto funciona bien
  concentrador para lo general
  emparejamiento directo para los pares con MUCHO tráfico
  → evita pagar el procesamiento por gigabyte dos veces
```

Lo que un concentrador aporta, además de reducir conexiones:

```
un punto donde ver el tráfico entre redes
un punto donde aplicar inspección                clase 200
segmentación por tabla de rutas, no por reglas
conexión con los enlaces dedicados en un solo sitio
                                                 clase 198
y conectividad entre regiones sin emparejar todo con todo
```

2. Segmentar con tablas, no con reglas

El error más común al montar un concentrador es asociarlo todo a una sola tabla de rutas. Entonces todo alcanza a todo, y la segmentación queda dependiendo de reglas de cortafuegos en cada extremo.

```
UNA SOLA TABLA
  producción, desarrollo, socios y oficinas se ven entre sí
  → un fallo de configuración en cualquier extremo abre un
    camino                                          clase 189
  → y el alcance desde un punto comprometido es total

VARIAS TABLAS
  cada red se ASOCIA a una tabla (qué rutas ve)
  y PROPAGA sus rutas a las tablas que decidas (quién la ve)
  → asociación y propagación son cosas distintas
```

Y el patrón que resuelve la mayoría de los casos:

```
TABLA producción
  asocia    redes de producción
  ve        producción + servicios compartidos + corporativa
  NO ve     desarrollo, preproducción, socios

TABLA no-producción
  asocia    desarrollo, preproducción
  ve        no-producción + servicios compartidos
  NO ve     producción              ← control estructural

TABLA compartidos
  asocia    servicios comunes (nombres, identidad, registro)
  ve        todos

TABLA socios
  asocia    conexiones de terceros
  ve        solo la red de intercambio
  → un socio no alcanza nada más, por construcción
```

Y la propiedad que hace esto valioso:

```
«preproducción no puede alcanzar producción» deja de ser una
regla que alguien puede quitar
→ pasa a ser una ausencia de ruta
→ y las ausencias no se saltan por error de configuración
                                                    clase 189
```

La inspección centralizada, cuando hace falta:

```
si el tráfico entre segmentos debe inspeccionarse
  las tablas apuntan al cortafuegos, no a la red destino
  el cortafuegos decide y devuelve al concentrador

coste
  un salto más: +0,3 a 1 ms
  el gigabyte se procesa DOS veces en el concentrador
  y el cortafuegos hay que dimensionarlo y hacerlo redundante

→ inspecciona lo que cruza fronteras de confianza, no todo
                                                    clase 189
```

Y un fallo de diseño frecuente:

```
hacer pasar por el cortafuegos central el tráfico entre dos
redes del MISMO segmento
→ triplica el coste y añade latencia sin ganar nada
```

3. Lo que cuesta

Un concentrador tiene dos costes que se suman y uno que no se ve.

```
POR CONEXIÓN Y HORA
  cada red conectada paga una tarifa fija
  → con 60 redes es una cifra apreciable al mes

POR GIGABYTE PROCESADO
  se cobra el tráfico que pasa por el concentrador
  → y se paga en CADA salto
```

```
Y EL QUE NO SE VE
  el tráfico que antes iba directo y ahora da un rodeo
  → dos redes en la misma zona que se hablaban directamente
    ahora pasan por el concentrador: se paga el proceso y
    se añade latencia
```

Y el cálculo que hay que hacer antes:

```
por cada par de redes con tráfico importante
  volumen mensual × precio por GB × número de saltos

y compararlo con
  el coste de mantener un emparejamiento directo (casi cero)

→ los pares de mucho volumen suelen merecer emparejamiento
  directo ADEMÁS del concentrador
→ pero cuidado: entonces hay dos caminos, y hay que evitar
  la asimetría                                    clase 194
```

La latencia, que es pequeña pero se acumula:

```
emparejamiento directo, misma zona          ~0,2 ms
concentrador                                 +0,3-1 ms
concentrador + inspección                    +1-3 ms
concentrador entre regiones                 el de la región

→ irrelevante para casi todo
→ importante si el camino crítico cruza varias veces
  clase 186
```

Multirregión y multinube, que es donde el concentrador aporta más:

```
CONCENTRADORES EMPAREJADOS ENTRE REGIONES
  un concentrador por región, conectados entre sí
  → cada red se conecta solo a su concentrador local
  → y alcanza la otra región sin emparejar nada más

MULTINUBE
  cada nube tiene su propio concentrador
  la unión se hace con enlaces dedicados o túneles
                                                clase 198
  → y el plan de direccionamiento tiene que estar limpio,
    o nada de esto funciona                     clase 193

Y LA REGLA QUE EVITA EL DESASTRE
  no encadenes tránsito entre nubes sin decidirlo
  → «la nube A alcanza la C a través de la B» suele ser un
    accidente de propagación, no una decisión
```

Y una consecuencia de coste que sorprende:

```
el tráfico entre nubes atraviesa dos concentradores y una
salida
→ se paga proceso en los dos y salida en el origen
→ por eso el tráfico entre nubes conviene medirlo por
  separado y atribuirlo                         clase 168
```

4. Cuándo NO usar un concentrador

Un concentrador no siempre es la respuesta, y montarlo por defecto tiene costes.

```
NO LO USES SI
  hay menos de 6-8 redes y no se prevé crecer
  el tráfico entre dos redes es enorme y directo
    → empareja esas dos y deja el resto
  el objetivo era aislar y se va a asociar todo a una tabla
    → entonces solo has creado un punto central sin ganar
      aislamiento
  no hay quien lo opere
    → un concentrador mal operado es un punto único   ley 20
```

Y el riesgo estructural que introduce:

```
UN CONCENTRADOR ES UN PUNTO POR EL QUE PASA CUANTO SE MUEVE
  su fallo afecta a todas las redes conectadas
  un cambio de tabla erróneo puede aislar medio sistema
  y su límite de rutas o de ancho de banda es compartido
```

```
→ los proveedores lo hacen redundante por dentro, pero el
  error de configuración no lo cubre nadie
→ los cambios se hacen con ventana y vuelta atrás  clase 194
```

La operación, que tiene sus propias señales:

```
rutas por tabla, contra el límite                ley 13
tráfico procesado por conexión y su coste
conexiones en estado no disponible
cambios en asociaciones y propagaciones ← auditar siempre
latencia entre pares representativos
```

Y la alerta más útil:

```
«ha aparecido una ruta hacia un segmento que no debería
 verse desde aquí»
→ detecta la propagación accidental, que es como se abren
  los caminos indeseados                          clase 189
```

Y la lista de comprobación de la clase:

- el número de redes justifica el concentrador
- hay varias tablas de rutas, no una
- asociación y propagación están decididas por segmento
- producción no es alcanzable desde no-producción por ausencia de ruta
- los socios solo ven la red de intercambio
- la inspección se aplica a lo que cruza fronteras, no a todo
- los pares de mucho volumen se han evaluado para emparejamiento directo
- está calculado el coste por gigabyte y por salto
- no hay tránsito accidental entre nubes
- hay alerta de rutas por tabla contra el límite
- hay alerta de rutas inesperadas en una tabla
- los cambios de tabla se hacen con ventana y vuelta atrás

Y el cierre que enlaza con la clase siguiente: el concentrador resuelve la conectividad entre redes propias, pero el tráfico hacia los servicios del proveedor y hacia internet sigue saliendo por otro lado. Puntos privados y control de salida es la materia de la clase 200.

Ejemplo trabajado

CloudShop tiene 63 redes en tres nubes, conectadas con 214 emparejamientos directos acumulados en cinco años. Lo que sigue es la migración a concentradores, la segmentación que cerró un camino que nadie sabía que existía, y los dos pares que se dejaron emparejados a propósito.

El punto de partida:

```
redes                                              63
emparejamientos directos                          214
  documentados                                     47
  creados «para una prueba» y nunca retirados      31
  sin dueño identificable                          52
rutas en la tabla de la red de producción         187
límite de la nube                                 200
→ a 13 rutas de no poder añadir una red más

tiempo medio para conectar una red nueva      6 días
  → y por eso 31 emparejamientos se hicieron «rápido»
                                                  ley 16
```

Y el hallazgo del inventario, que fue el que justificó el proyecto:

```
se trazaron los caminos alcanzables desde cada red

  desde la red de DESARROLLO se alcanzaba
    la red de datos de PRODUCCIÓN

  cómo
    desarrollo ↔ herramientas (emparejado en 2021)
    herramientas ↔ datos-producción (emparejado en 2022)
    → y alguien había añadido rutas estáticas en ambos
      extremos «para una migración» que terminó en 2022

  llevaba así                                 3 años
```

```
  aparecía en algún diagrama                  no    ley 24
  lo detectó                    el inventario, no una alerta
```

La segmentación diseñada.

```
TABLA producción
  asocia      18 redes de producción
  propaga a   producción, compartidos
  ve          producción, compartidos, corporativa

TABLA no-producción
  asocia      29 redes (desarrollo, preproducción, pruebas)
  propaga a   no-producción, compartidos
  ve          no-producción, compartidos
  NO ve       producción            ← ausencia de ruta

TABLA compartidos
  asocia      7 redes (nombres, identidad, registro,
              artefactos, telemetría)
  propaga a   todas
  ve          todas

TABLA socios
  asocia      4 conexiones de terceros
  propaga a   solo intercambio
  ve          solo la red de intercambio

TABLA inspección
  el tráfico producción ↔ corporativa y todo lo de socios
  pasa por el cortafuegos central
```

Y la comprobación del diseño, hecha antes de migrar:

```
se simuló, tabla por tabla, qué alcanza cada red

  desde desarrollo
    compartidos                                 sí
    otras redes de desarrollo                   sí
    producción                                  NO
    datos de producción                         NO
    corporativa                                 NO

  desde una red de socio
    red de intercambio                          sí
    cualquier otra cosa                         NO
```

La migración, por lotes.

```
semana 1     concentrador en la región principal; tablas
             creadas y vacías
semanas 2-4  redes de desarrollo, 29, de 6 en 6
             y retirada del emparejamiento correspondiente
             en cada paso                       clase 184
semanas 5-8  redes de producción, 18, de 3 en 3, en ventana
semanas 6-9  compartidos y socios
semana 10    concentrador de la segunda región, emparejado
semanas 11-13 las otras dos nubes, por enlace dedicado
             clase 198
semanas 12-15 RETIRADA de los 214 emparejamientos
```

Y el paso de retirada, que es el que casi siempre falta:

```
emparejamientos retirados                       212
conservados a propósito                           2   ← ver abajo

y durante la retirada
  4 emparejamientos resultaron estar en uso por algo que
  nadie había declarado
    · un trabajo de exportación nocturno
    · una herramienta de monitorización de un proveedor
    · un servidor de compilación antiguo
    · una réplica de base de datos de un proyecto cancelado
      que seguía copiando datos       ← se apagó, ley 20
```

Los dos emparejamientos que se conservaron, con su cálculo:

```
PAR 1   aplicación ↔ base de datos de pedidos
  volumen mensual                          412 TB
  por el concentrador
    412.000 GB × 0,02 €/GB                8.240 €/mes
    latencia añadida                      +0,6 ms en el
                                          camino crítico
  emparejado directo
    coste de proceso                          0 €
    latencia                              0,2 ms
  decisión   emparejamiento directo, y ruta más específica
             para que gane al camino del concentrador
                                                 clase 194
  ahorro     8.240 €/mes

PAR 2   telemetría ↔ recolector central
  volumen mensual                          180 TB
  ahorro                                3.600 €/mes
  decisión   igual
```

Y la precaución que hizo falta:

```
al existir dos caminos posibles, hubo que comprobar simetría
  la ruta más específica gana en las DOS direcciones
  se verificó con una traza en ambos sentidos     clase 194
  y se añadió a la comprobación semanal
```

El coste, antes y después:

```
emparejamientos                              214           2
rutas en la tabla de producción              187          23
conexiones al concentrador                     0          63
coste fijo del concentrador                    0     2.840 €/mes
coste por GB procesado                         0     6.100 €/mes
  (tras excluir los 2 pares directos)
coste evitado por los 2 pares directos         —    11.840 €/mes
tiempo para conectar una red nueva          6 días      25 min
caminos indeseados detectables                 no          sí
```

Y la línea que resume el balance:

```
el concentrador cuesta 8.940 €/mes
sin los dos emparejamientos directos costaría 20.780 €
→ la decisión de dejar dos pares fuera del concentrador
  valía más que todo el resto del proyecto en dinero
```

La vigilancia montada:

```
rutas por tabla contra el límite, alerta al 75 %
cambios en asociaciones y propagaciones → auditados y
  con alerta
RUTA INESPERADA EN UNA TABLA
  se declara qué prefijos deben verse desde cada tabla
  cualquier otro dispara alerta
  → en el primer trimestre disparó 2 veces
    · una propagación mal configurada al añadir una red
    · un socio nuevo asociado a la tabla equivocada
  → las dos se corrigieron en minutos, no en años
```

La lección que esta clase deja: el proyecto se justificó por el límite de rutas, pero lo que encontró fue un camino de desarrollo a los datos de producción que llevaba tres años abierto a través de dos emparejamientos y unas rutas estáticas de una migración terminada. Ninguna regla de cortafuegos lo habría cerrado de forma duradera; lo cerró la ausencia de ruta. Y en dinero, la decisión que más valió fue la contraria a la del proyecto: dejar dos pares fuera del concentrador, porque su volumen hacía que el coste por gigabyte superara todo lo demás.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-16-advanced-cloud-networking-edge/199-transit-gateway-virtual-wan-y-network-
connectivity-center/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 transit-comparison 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 transit-comparison 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
Añadir una red nueva exige tocar decenas de tablas	Topología de emparejamientos punto a punto, que crece de forma cuadrática	Migra a un concentrador cuando pases de seis u ocho redes, y retira los emparejamientos a medida que migras.
El concentrador no aporta aislamiento	Todas las redes están asociadas a una única tabla de rutas	Usa varias tablas y decide por separado a qué tabla se asocia cada red y a cuáles propaga sus rutas.
Un entorno inferior alcanza producción y nadie sabe por dónde	Tránsito accidental a través de emparejamientos encadenados y rutas estáticas olvidadas	Traza qué alcanza cada red, cierra por ausencia de ruta en vez de por regla y alerta ante rutas inesperadas en una tabla.
La factura del concentrador es enorme	Pares con muchísimo volumen atraviesan el concentrador y pagan proceso por gigabyte	Calcula el coste por par y deja emparejamiento directo para los de gran volumen, cuidando la simetría del camino.
La latencia del camino crítico empeora tras centralizar	Tráfico que antes iba directo ahora da un rodeo, a veces con inspección	Inspecciona solo lo que cruza fronteras de confianza y evita enrutar por el cortafuegos el tráfico dentro de un mismo segmento.
Un cambio de tabla aísla medio sistema	El concentrador es un punto por el que pasa todo y el cambio se hizo sin red	Cambios en ventana, con configuración previa guardada y vuelta atrás probada, y auditoría de asociaciones y propagaciones.

■ Seguridad, ética y costo

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

■ Preguntas de comprobación

1. ¿Por qué el emparejamiento directo deja de escalar y a partir de cuántas redes?
2. ¿Qué diferencia hay entre asociar una red a una tabla y propagar sus rutas a ella?
3. ¿Por qué cerrar un camino por ausencia de ruta es más robusto que por regla?
4. ¿Qué costes tiene un concentrador y cuál es el que no se ve?
5. ¿En qué casos conviene mantener un emparejamiento directo pese a tener concentrador?

Referencias

- AWS (2025). Transit Gateway route tables and segmentation. <https://docs.aws.amazon.com/vpc/latest/tgw/tgw-route-tables.html>
- Microsoft (2025). Virtual WAN architecture and routing. <https://learn.microsoft.com/en-us/azure/virtual-wan/virtual-wan-about>
- Google Cloud (2025). Network Connectivity Center. <https://cloud.google.com/network-connectivity/docs/network-connectivity-center>
- AWS (2025). Building a scalable and secure multi-VPC network infrastructure. <https://docs.aws.amazon.com/whitepapers/latest/building-scalable-secure-multi-vpc-network-infrastructure/welcome.html>
- Microsoft (2025). Cloud Adoption Framework: network topology and connectivity. <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/landing-zone/design-area/network-topology-and-connectivity>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 16 en PDF · Manual integral

Anterior	Índice	Siguiente
← 198 · VPN, Direct Connect, ExpressRoute e Interconnect	Parte 16 · Programa	200 · Private endpoints, service networking y egress control →

Evaluación — 199 Transit Gateway, Virtual WAN y Network Connectivity Center

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 transit-comparison 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.

200 — Private endpoints, service networking y egress control

Parte: 16 — Redes cloud avanzadas, conectividad híbrida y edge Nivel: avanzado · Horas estimadas: 4 Laboratorio: security · Estado: EXECUTABLECORE

Propósito

Cerrar el camino de salida, que es la mitad que casi nadie controla: se filtran las entradas con cuidado y el tráfico saliente sale por donde quiere. La clase explica los puntos privados hacia los servicios del proveedor —qué resuelven, qué cuestan y qué no cubren—, y desarrolla el control de salida como lo que de verdad es: la última barrera contra la exfiltración, y la que más veces aparece como prueba negativa fallida en todo este programa.

Resultados de aprendizaje

Al finalizar podrás:

1. Alcanzar servicios del proveedor sin pasar por internet, con el mecanismo adecuado.
2. Distinguir punto privado de punto de servicio y saber qué protege cada uno.
3. Cerrar la salida por defecto y abrirla por destino declarado.
4. Detectar los caminos de exfiltración que quedan abiertos.
5. Medir lo que el control de salida ahorra en coste, además de en riesgo.

Conceptos centrales

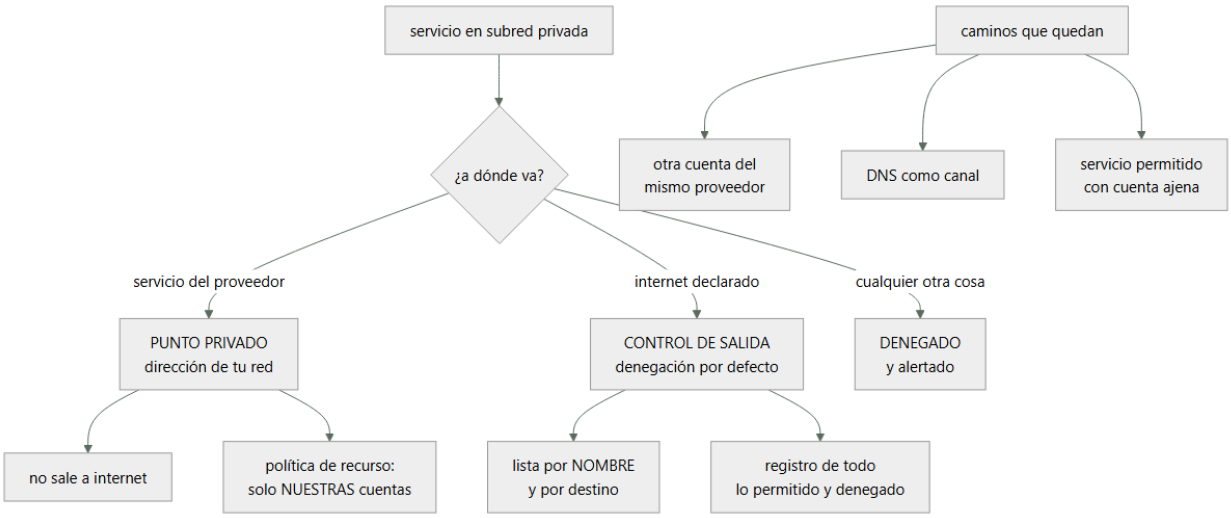
Concepto	Comprensión verificable
punto privado	Interfaz con dirección de tu red que representa un servicio del proveedor. El tráfico no sale a internet.
punto de servicio	Ruta hacia el servicio del proveedor sin dirección propia. Más barato y con menos control de origen.
control de salida	Filtrado del tráfico que sale de la red, por destino y por nombre, con denegación por defecto.
exfiltración	Sacar datos a un destino no autorizado. La salida abierta es su camino natural.
política de perímetro de datos	Regla que impide que un recurso sea alcanzado desde fuera de la organización o alcance recursos ajenos.
inspección con nombre	Filtrado por el nombre solicitado y no solo por dirección, necesario porque las direcciones de los servicios cambian.

Modelo mental

La red cloud es un sistema distribuido de rutas, identidades y políticas; cada salto añade latencia, costo y una frontera de fallo.

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. Alcanzar servicios sin salir a internet

Un servicio en una subred privada que llama al almacén de objetos del proveedor sale, por defecto, a internet: hasta la pasarela NAT, hasta el punto público del servicio, y de vuelta.

```
QUÉ TIENE DE MALO
  coste de salida por cada gigabyte              clase 168
  el tráfico atraviesa internet, aunque vaya cifrado
  la subred necesita ruta a internet, y esa ruta sirve
    también para sacar datos a cualquier otro sitio
  la pasarela NAT es un cuello y un coste por hora y por GB
```

Las dos soluciones, que no son equivalentes:

```
PUNTO DE SERVICIO (ruta privilegiada)
  añade una ruta hacia el servicio; el tráfico no sale a
  internet
  la dirección de origen sigue siendo de la red
  + barato o gratuito, simple
  – no da una dirección propia; no se puede alcanzar desde
    la corporativa ni desde otra red
  – el filtrado por origen es más grosero

PUNTO PRIVADO (interfaz en tu red)
  crea una interfaz CON DIRECCIÓN de tu subred que representa
  al servicio
  + alcanzable desde la corporativa y desde otras redes
  + permite política por origen fina
  + resuelve por nombre a una dirección privada  clase 195
  – cuesta por hora y por gigabyte
  – uno por servicio y por zona → consume direcciones
                                                 clase 193
```

Y el criterio de elección:

```
¿hace falta alcanzarlo desde fuera de esta red?
  sí → punto privado
  no → punto de servicio, si el proveedor lo ofrece

¿el volumen es enorme?
  compara el coste por GB del punto privado contra el de
  salida a internet; casi siempre gana el punto privado,
  pero no siempre
```

Y lo que un punto privado no resuelve, que es lo que más se malinterpreta:

```
un punto privado te lleva al SERVICIO, no a TU recurso
→ desde ese punto se puede llegar al almacén de otra
  organización si nada lo impide
→ y por tanto sigue siendo un camino de exfiltración
```

```
LO QUE LO CIERRA
  política en el punto privado: solo hacia recursos de
    NUESTRAS cuentas
  política en el recurso: solo desde NUESTRAS redes
  → las dos direcciones, no una                  clase 189
```

2. Cerrar la salida

La entrada se filtra con cuidado desde siempre. La salida, casi nunca, y es por donde se van los datos.

```
EL ESTADO HABITUAL
  la subred de aplicación tiene ruta 0.0.0.0/0
  el grupo de seguridad permite todo el tráfico saliente
  → un servicio comprometido puede enviar lo que quiera a
    donde quiera
  → y en este programa, la prueba negativa «sacar datos a
    un destino externo no declarado» ha fallado tres veces
                                       clases 179, 189, 194
```

El diseño correcto, en capas:

```
1  SIN RUTA A INTERNET donde no haga falta
   la subred de datos no necesita salir             clase 194
   → es el control más fuerte: ausencia de camino

2  SALIDA CENTRALIZADA
   0.0.0.0/0 hacia un cortafuegos de salida, no hacia la
   pasarela NAT directamente
   → un solo punto donde ver y decidir

3  DENEGACIÓN POR DEFECTO, CON LISTA POR NOMBRE
   se permite lo declarado: repositorios de paquetes,
   proveedores concretos, servicios de terceros
   → por NOMBRE, porque las direcciones de esos servicios
     cambian constantemente

4  REGISTRO DE LO PERMITIDO Y DE LO DENEGADO
   → lo denegado es la señal que descubre lo que no sabías

5  PERÍMETRO DE DATOS
   políticas que impiden que un recurso propio sea alcanzado
   por identidades ajenas, y que una identidad propia alcance
   recursos ajenos
```

Y la parte 5 merece detalle porque es lo que cierra el hueco del punto privado:

```
DOS REGLAS, EN LAS DOS DIRECCIONES
  «ninguna identidad de fuera de la organización puede
   acceder a nuestros recursos»
  «ninguna identidad nuestra puede acceder a recursos que no
   son de la organización»

→ la segunda es la que impide copiar datos al almacén de
  otro, y es la que casi nunca está
```

Los caminos que quedan abiertos aunque se cierre lo evidente:

```
OTRA CUENTA DEL MISMO PROVEEDOR
  el punto privado alcanza el servicio, y el servicio
  alcanza cualquier cuenta
  → lo cierra el perímetro de datos, no el cortafuegos

DNS COMO CANAL
  las consultas de nombres salen aunque el resto esté
  cerrado, y se pueden usar para sacar datos poco a poco
  → obliga a que la resolución pase por el resolutor propio
    y a vigilar consultas anómalas                clase 195

UN SERVICIO PERMITIDO USADO CON OTRA CUENTA
  si se permite un servicio de almacenamiento por nombre,
  se permite también la cuenta de un atacante en ese mismo
  servicio
  → hace falta filtrar por recurso, no solo por nombre de
    servicio

CANALES DE ACTUALIZACIÓN Y TELEMETRÍA
```

```
  agentes que envían datos a un tercero por diseño
  → hay que saber qué envían                          ley 20
```

3. Lo que cuesta y lo que ahorra

Cerrar la salida suele presentarse como coste de seguridad, y en la práctica ahorra dinero.

```
LO QUE SE AHORRA
  el tráfico hacia los servicios del proveedor deja de
  pagar salida a internet
  la pasarela NAT procesa mucho menos: menos coste por hora
  y por gigabyte
  y aparecen destinos que nadie sabía que se estaban
  pagando

LO QUE CUESTA
  puntos privados: por hora y por gigabyte
  cortafuegos de salida: instancia y proceso
  el trabajo de descubrir qué hace falta permitir
```

Y el orden de implantación que funciona, porque el error de cerrar de golpe es romper producción:

```
1  REGISTRAR SIN BLOQUEAR, dos o cuatro semanas
   → produce la lista real de destinos, que nunca coincide
     con la que la gente cree

2  CLASIFICAR los destinos observados
   necesarios · innecesarios · desconocidos
   → los desconocidos son el hallazgo

3  PERMITIR lo necesario, por nombre

4  BLOQUEAR en un entorno inferior y observar qué se rompe

5  BLOQUEAR en producción, empezando por las subredes que
   menos salen

6  DEJAR EL REGISTRO de denegaciones vigilado    ley 13
   → cada denegación nueva es información
```

Y una advertencia sobre el paso 3:

```
si permitir un destino nuevo tarda dos días, alguien pedirá
abrir «*.amazonaws.com» y se acabó el control    ley 16
→ el camino para pedir una excepción tiene que ser rápido,
  con dueño y caducidad                          clase 190
```

Lo que hay que vigilar, una vez cerrado:

```
denegaciones por destino y por origen
destinos permitidos que dejan de usarse    → retirar
volumen saliente por servicio y por destino  ← el que sube
  sin motivo es la señal de exfiltración
consultas de nombres anómalas: muchas, largas o a dominios
  recién registrados
tráfico por la pasarela NAT: debería BAJAR mucho
```

Y la prueba negativa que cierra la clase, que es la que este programa ha visto fallar más veces:

```
desde un servicio en la subred de aplicación, intentar
  subir un fichero a un almacén de otra organización
  enviar datos a un servidor externo por HTTPS
  sacar datos por consultas de nombres
  usar un servicio permitido con credenciales ajenas
→ las cuatro deben fallar y quedar registradas    ley 22
```

4. Puntos privados en la práctica

Montar puntos privados a escala tiene detalles que dan problemas si se descubren tarde.

```
CONSUMO DE DIRECCIONES
  uno por servicio y por zona
  30 servicios × 3 zonas = 90 direcciones
  → hay que contarlo en el plan                    clase 193

RESOLUCIÓN
```

```
  el nombre público del servicio debe resolver a la dirección
  privada DENTRO de la red
  → zona privada asociada, y reenvío desde la corporativa
    si hace falta                                  clase 195
  → si esto falla, el tráfico sigue saliendo a internet sin
    que nadie lo note

COSTE
  por hora y por gigabyte procesado
  → con muchos servicios y poco tráfico, el coste fijo
    domina
  → conviene compartir puntos privados entre redes cuando el
    proveedor lo permita

CENTRALIZACIÓN
  un conjunto de puntos privados en una red compartida,
  alcanzable desde el concentrador               clase 199
  → evita replicarlos en 60 redes
  → pero concentra el tráfico y su coste
```

Y la comprobación que evita el error silencioso:

```
desde dentro de la red, resolver el nombre del servicio
  → debe devolver una dirección PRIVADA
  → si devuelve una pública, el punto privado existe y no
    se está usando: se paga por él y el tráfico sale igual

→ función de aptitud: para cada servicio con punto privado,
  comprobar que resuelve a privada desde cada red
                                                 clase 190
```

Y la lista de comprobación de la clase:

- las subredes que no deben salir no tienen ruta a internet
- la salida pasa por un cortafuegos con denegación por defecto
- la lista de destinos permitidos es por nombre y está justificada
- hay perímetro de datos en las dos direcciones
- los puntos privados tienen política de recurso restringida
- los nombres de servicio resuelven a direcciones privadas desde cada red
- la resolución de nombres pasa por el resolutor propio
- se registran las denegaciones y alguien las mira
- pedir una excepción es rápido y deja dueño y caducidad
- se vigila el volumen saliente por destino
- las cuatro pruebas negativas de exfiltración se han ejecutado y fallan como deben
- se ha medido la reducción de coste de salida y de NAT

Y el cierre que enlaza con la clase siguiente: cerrada la salida hacia fuera, queda el tráfico entre servicios dentro del sistema, que en la mayoría de las redes sigue siendo abierto y sin identidad. Malla de servicios, TLS mutuo y gestión del tráfico este-oeste es la materia de la clase 201.

Ejemplo trabajado

CloudShop cierra la salida de sus 63 redes. Lo que sigue es lo que apareció en las cuatro semanas de registro sin bloquear, los tres caminos de exfiltración que quedaban abiertos tras el primer cierre, y el ahorro que nadie esperaba.

Semanas 1-4: registrar sin bloquear.

```
destinos externos observados, únicos                1.847

clasificación
  necesarios y conocidos                              61
    repositorios de paquetes, proveedores de pago,
    servicios de correo, actualizaciones de sistema
  del propio proveedor de nube                       340
    → deberían ir por punto privado, no por internet
  innecesarios                                       112
    servicios que se dejaron de usar y cuyos clientes
    seguían intentando conectar
  DESCONOCIDOS                                     1.334
```

Y el análisis de los desconocidos, que es donde estaba la información:

```
1.291  dominios de publicidad y analítica alcanzados por
       una biblioteca de una aplicación interna
       → la biblioteca cargaba recursos de terceros en un
         panel de administración

   28  servicios de un proveedor de monitorización
       contratado por un equipo en 2022 y nunca dado de
       baja; el agente seguía enviando métricas       ley 20
       → incluidas métricas con nombres de tablas y de
         clientes

    9  un servicio de traducción automática al que un
       microservicio enviaba textos de descripciones de
       producto
       → no estaba declarado en ningún sitio; lo añadió un
         desarrollador en 2023 y funcionaba

    4  almacenes de objetos de OTRAS organizaciones
       → tres eran de socios, con acuerdo
       → UNO era el almacén personal de un antiguo
         empleado, con un script de exportación que seguía
         ejecutándose desde una máquina de análisis
         → llevaba 14 meses copiando un extracto diario
           de ventas

    2  dominios registrados hacía menos de 30 días
       → resultaron ser de una herramienta legítima que
         había cambiado de dominio
```

Y la conclusión del registro:

```
el hallazgo más grave del proyecto apareció ANTES de
bloquear nada
→ registrar sin bloquear es la fase que más encuentra
→ y la exfiltración real llevaba 14 meses sin que ninguna
  alerta existiera                                  ley 15
```

El cierre, por fases.

```
fase 1   subredes de datos: se les quitó la ruta a internet
         → 0 destinos externos legítimos; cierre limpio
         → y el script de exportación del antiguo empleado
           dejó de funcionar aquí

fase 2   340 destinos del proveedor → puntos privados
         puntos privados creados                      27
         centralizados en una red compartida, alcanzables
         desde el concentrador                clase 199
         direcciones consumidas                       81

fase 3   subredes de aplicación: cortafuegos de salida con
         denegación por defecto
         destinos permitidos, por nombre              61
         → despliegue primero en preproducción: 4 cosas se
           rompieron, todas destinos legítimos no
           declarados
         → en producción, 1 más

fase 4   perímetro de datos, en las dos direcciones
```

Los tres caminos que quedaban abiertos tras la fase 3. Se descubrieron ejecutando las pruebas negativas:

```
PRUEBA 1   subir un fichero al almacén de otra organización
  resultado   FUNCIONÓ
  por qué     el punto privado del almacén permitía llegar
              al servicio, y el servicio a cualquier cuenta
  corrección  política en el punto privado: solo recursos
              de nuestras cuentas
              + política en los recursos: solo desde
              nuestras redes

PRUEBA 2   sacar datos por consultas de nombres
  resultado   FUNCIONÓ
```

```
  por qué     las consultas salían al resolutor por defecto
              sin restricción; se pudieron sacar 40 KB en
              consultas codificadas
  corrección  toda resolución pasa por el resolutor propio
              alerta por consultas anómalas: longitud,
              volumen por origen, dominios recién
              registrados

PRUEBA 3   usar un servicio permitido con credenciales de
           otra cuenta
  resultado   FUNCIONÓ parcialmente
  por qué     el filtro era por nombre de servicio, no por
              recurso
  corrección  filtrado por recurso donde el producto lo
              permite; perímetro de datos donde no

PRUEBA 4   enviar datos a un servidor externo por HTTPS
  resultado   denegado y registrado                    ✓
```

Y la observación del equipo:

```
tres de las cuatro pruebas fallaron después de haber
cerrado la salida
→ y las tres se habrían dado por resueltas leyendo la
  configuración                                    ley 22
```

El coste, antes y después:

```
tráfico por pasarelas NAT             71 TB/mes     9 TB/mes
coste de pasarelas NAT (proceso)       3.550 €      450 €
coste de salida a internet             5.900 €    1.100 €
coste de puntos privados                     0    1.780 €
coste del cortafuegos de salida              0      920 €
████████████████████████████████████████████████████████████
total                                  9.450 €    4.250 €

ahorro                                          5.200 €/mes
```

Y la línea que sorprendió a finanzas:

```
cerrar la salida AHORRÓ dinero
→ porque el 82 % del tráfico saliente iba a servicios del
  propio proveedor y se estaba pagando como salida a
  internet                                       clase 168
→ y ese gasto no tenía dueño ni atribución          ley 20
```

La operación posterior, a los seis meses:

```
denegaciones registradas                      1.400/mes
  de ellas, destinos legítimos no declarados            7
    → los 7 se añadieron; tiempo medio de aprobación 40 min
  de ellas, clientes de servicios ya retirados      1.290
    → se corrigieron 4 servicios; el ruido bajó a 110/mes
  de ellas, intentos que valía la pena investigar       3
    · un agente antiguo reinstalado por una imagen vieja
    · una biblioteca nueva que llamaba a un servicio de
      telemetría propio
    · un script de un becario apuntando a un servicio
      de conversión de ficheros externo

destinos permitidos que dejaron de usarse            11
  → retirados; la lista pasó de 68 a 57

puntos privados con resolución incorrecta             2
  → la función de aptitud los detectó: existían, se
    pagaban, y el tráfico salía igual por internet
  → 340 €/mes que se pagaban dos veces
```

El resultado, con la evidencia:

```
destinos externos alcanzables           ilimitado         57
pruebas de exfiltración que fallan            3 de 4    0 de 4
coste mensual de conectividad saliente      9.450 €    4.250 €
tiempo para autorizar un destino nuevo        n/a       40 min
exfiltraciones activas detectadas               1           0
destinos con dueño declarado                    0          57
```

La lección que esta clase deja: el hallazgo más grave —un extracto diario de ventas que llevaba catorce meses copiándose al almacén personal de un antiguo empleado— apareció en la fase de registrar sin bloquear, antes de cerrar nada. Y tras cerrar la salida, tres de las cuatro pruebas de exfiltración seguían funcionando, porque un punto privado lleva al servicio y no a tu recurso. Cerrar la salida, además, salió 5.200 € al mes más barato que dejarla abierta.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-16-advanced-cloud-networking-edge/200-private-endpoints-service-networking-y-egress-control/lab.py
```

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

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

Evidencia esperada

El artefacto principal es un control con amenaza, mitigación y evidencia. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye private-connectivity 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 punto privado está montado y el tráfico sigue saliendo por internet	El nombre del servicio no resuelve a la dirección privada desde esa red	Asocia la zona privada correspondiente y comprueba con una función de aptitud que cada servicio resuelve a dirección privada desde cada red.
Se puede subir datos al almacén de otra organización pese a tener punto privado	El punto privado lleva al servicio, no a tus recursos	Aplica política en el punto privado hacia recursos propios y política en los recursos hacia redes propias; perímetro de datos en las dos direcciones.
Se sacan datos aunque el cortafuegos de salida esté cerrado	Las consultas de nombres salen sin restricción y sirven de canal	Fuerza la resolución por el resolutor propio y vigila consultas anómalas por longitud, volumen y antigüedad del dominio.
Alguien pide abrir un comodín enorme y se acaba el control	Autorizar un destino nuevo tarda días	Haz que pedir una excepción sea rápido y deje dueño y fecha de caducidad registrados.
Cerrar la salida rompe producción	Se bloqueó sin conocer los destinos reales	Registra sin bloquear varias semanas, clasifica lo observado, permite lo necesario y bloquea primero en entornos inferiores.
El control de salida se percibe como un gasto de seguridad	No se midió el tráfico que iba a servicios del propio proveedor pagando salida a internet	Mide antes y después el tráfico por pasarela NAT y el coste de salida; suele bajar más de lo que cuestan los puntos privados.

Seguridad, ética y costo

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

Preguntas de comprobación

1. ¿Qué diferencia hay entre un punto de servicio y un punto privado?
2. ¿Por qué un punto privado no impide por sí solo la exfiltración?
3. ¿En qué orden se implanta el control de salida sin romper producción?
4. ¿Qué tres caminos siguen abiertos tras cerrar el cortafuegos de salida?
5. ¿Por qué cerrar la salida suele ahorrar dinero?

Referencias

- AWS (2025). Building a data perimeter on AWS. <https://docs.aws.amazon.com/whitepapers/latest/building-a-data-perimeter-on-aws/building-a-data-perimeter-on-aws.html>
- AWS (2025). VPC endpoints and endpoint policies. <https://docs.aws.amazon.com/vpc/latest/privatelink/vpc-endpoints.html>
- Microsoft (2025). Azure Private Link and private endpoints. <https://learn.microsoft.com/en-us/azure/private-link/private-link-overview>
- Google Cloud (2025). VPC Service Controls — perímetros de servicio. <https://cloud.google.com/vpc-service-controls/docs/overview>
- MITRE ATT&CK (2025). Exfiltration over alternative protocol y DNS tunneling. <https://attack.mitre.org/tactics/TA0010/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Anterior	Índice	Siguiente
← 199 · Transit Gateway, Virtual WAN y Network Connectivity Center	Parte 16 · Programa	201 · Service mesh, mTLS y gestión de tráfico este-oeste →

Evaluación — 200 Private endpoints, service networking y egress control

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 private-connectivity 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.

201 — Service mesh, mTLS y gestión de tráfico este-oeste

Parte: 16 — Redes cloud avanzadas, conectividad híbrida y edge Nivel: avanzado · Horas estimadas: 4 Laboratorio: network · Estado: EXECUTABLECORE

Propósito

Decidir si hace falta una malla de servicios, y si hace falta, qué parte de ella. La clase separa las cinco capacidades que una malla ofrece y comprueba cuáles están ya resueltas en otra capa —que fue el hallazgo de la clase 152 y el de la hipótesis de la parte 16—, explica el TLS mutuo como identidad de carga y no como cifrado, y advierte de lo que una malla cuesta de verdad: latencia por salto, recursos por réplica y un plano de control más que operar.

Resultados de aprendizaje

Al finalizar podrás:

1. Enumerar las cinco capacidades de una malla y dónde más se resuelven.
2. Decidir si una malla compensa, con las capacidades que faltan.
3. Implantar identidad de carga con TLS mutuo y autorización por servicio.
4. Configurar reintentos, plazos y cortes sin empeorar la saturación.
5. Medir el coste real: latencia, recursos y operación.

Conceptos centrales

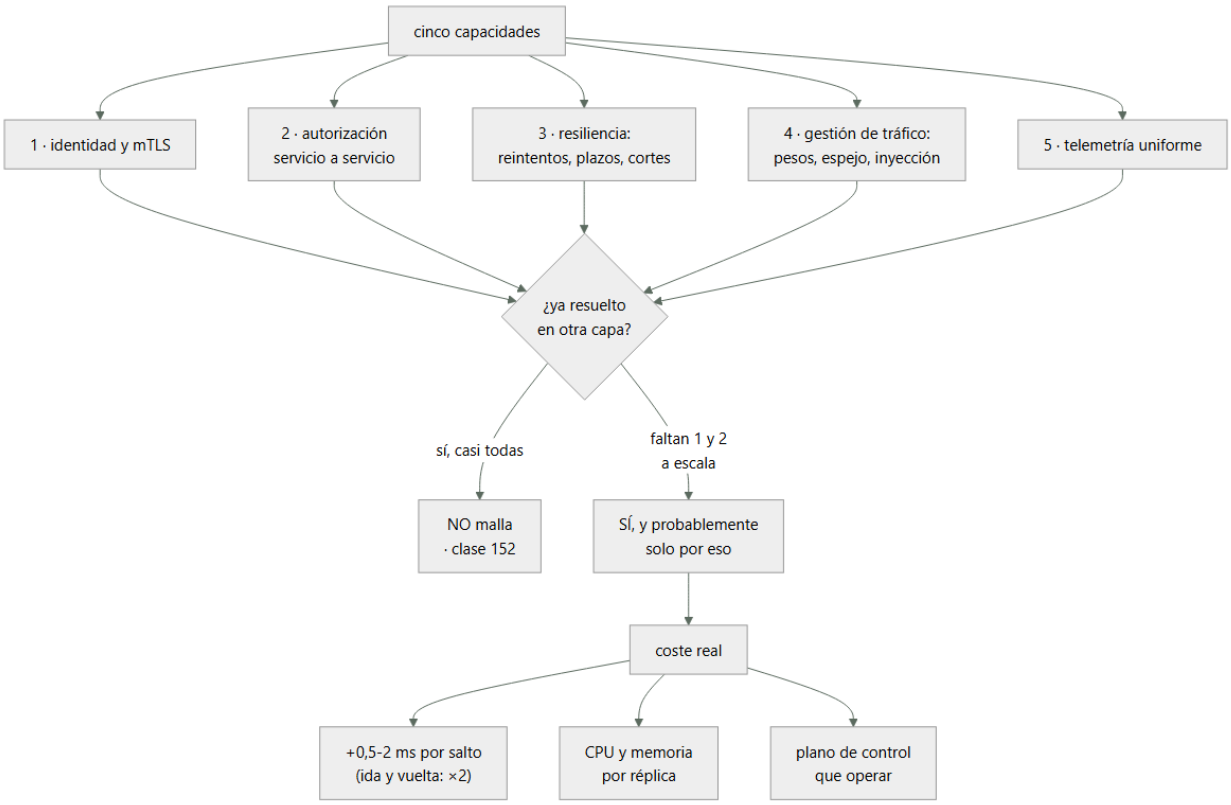
Concepto	Comprensión verificable
malla de servicios	Capa que intercepta el tráfico entre servicios para aplicar identidad, política, resiliencia y telemetría.
acompañante	Proceso o biblioteca que intercepta el tráfico de un servicio. Añade latencia y consume recursos por réplica.
TLS mutuo	Ambos extremos presentan certificado. Da identidad criptográfica a cada carga, sin credenciales guardadas.
autorización este-oeste	Regla que dice qué servicio puede llamar a qué servicio y con qué método.
plano de control	Componente que distribuye configuración a los acompañantes. Su fallo tiene consecuencias propias.
modo sin acompañante	Variante que evita un proceso por réplica usando el nodo o el propio protocolo. Reduce coste y limita capacidades.

Modelo mental

La red cloud es un sistema distribuido de rutas, identidades y políticas; cada salto añade latencia, costo y una frontera de fallo.

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. Las cinco capacidades, y dónde más se resuelven

Una malla se vende como un producto y en realidad son cinco cosas distintas. La pregunta útil no es «¿queremos malla?» sino «¿cuáles de las cinco nos faltan?».

```
1  IDENTIDAD Y CIFRADO ENTRE SERVICIOS (mTLS)
   dónde más se resuelve
     identidad de carga del proveedor + TLS en el balanceador
                                        clases 159, 196
     pero: eso cifra hasta el balanceador, no entre servicios
   → a escala, es la capacidad que MENOS alternativas tiene

2  AUTORIZACIÓN SERVICIO A SERVICIO
   dónde más se resuelve
     grupos de seguridad por servicio            clase 135
     comprobación en la aplicación
   → los grupos de seguridad autorizan por red, no por
     identidad; y con escalado dinámico se quedan cortos

3  RESILIENCIA: reintentos, plazos, cortes, mamparos
   dónde más se resuelve
     bibliotecas del lenguaje                    clase 153
     el balanceador de capa 7                    clase 196
   → si hay pocos lenguajes, la biblioteca basta
   → si hay siete, la malla evita implementarlo siete veces

4  GESTIÓN DE TRÁFICO: pesos, espejo, inyección de fallos
   dónde más se resuelve
     el balanceador de capa 7 y la canalización  clase 102
   → para el tráfico de entrada, el balanceador basta
   → para el tráfico interno entre servicios, no

5  TELEMETRÍA UNIFORME
   dónde más se resuelve
     instrumentación de la aplicación            clase 121
   → la malla da métricas de red por par de servicios sin
     tocar código, que es real y útil
   → pero NO da trazas de dentro de la aplicación
```

Y la conclusión del análisis, que coincide con lo que este programa ya encontró:

```
de las cinco, en un sistema con balanceador de capa 7 bien
configurado y pocos lenguajes
  1  falta                    ← esta sí
  2  falta a escala           ← esta también
  3  resuelta
  4  resuelta para lo que importa
  5  parcialmente resuelta

→ la malla se justifica por 1 y 2, o no se justifica
→ y si se adopta, conviene adoptar SOLO eso al principio
```

Y la advertencia que da esta clase:

```
adoptar una malla por las cinco capacidades cuando tres ya
están resueltas produce dos implementaciones de lo mismo
→ dos sitios donde configurar reintentos
→ y reintentos anidados, que multiplican la carga  ← ver abajo
```

2. Identidad de carga y autorización

TLS mutuo se explica mal como «cifrado entre servicios». Cifrar es lo de menos: lo importante es que cada carga tiene una identidad criptográfica que no se puede robar y copiar.

```
CÓMO FUNCIONA
  cada carga recibe un certificado de corta vida (horas)
  emitido por una autoridad interna, en función de su
  identidad de despliegue
  al conectar, ambos extremos presentan y verifican

QUÉ DA
  identidad sin credenciales guardadas            clase 159
  rotación automática y corta
  autorización basada en QUIÉN llama, no en desde qué IP
  y cifrado, de propina
```

Y por qué la identidad por red no basta a escala:

```
UN GRUPO DE SEGURIDAD DICE
  «desde esta subred se puede llegar al puerto 8080»

CON ESCALADO DINÁMICO Y CONTENEDORES
  en esa subred hay 40 servicios distintos
  → autorizar por red autoriza a los 40
  → y la dirección cambia cada pocos minutos

CON IDENTIDAD DE CARGA
  «el servicio pedidos puede llamar a POST /cobrar del
   servicio pagos»
  → y nada más
```

La autorización este-oeste, que es lo que convierte la identidad en control:

```
regla por origen, destino, método y ruta
  origen   pedidos
  destino  pagos
  permite  POST /cobrar, POST /reembolsar
  deniega  el resto

y con denegación por defecto
  → el alcance desde un servicio comprometido pasa de
    «todo lo de la red» a «tres llamadas»          clase 189
```

Y el orden de implantación que no rompe nada:

```
1  mTLS en modo PERMISIVO
   se acepta tráfico cifrado y sin cifrar
   → nada se rompe; se empieza a ver quién habla con quién

2  observar el grafo real de llamadas
   → y comprobarlo contra el diagrama              ley 24

3  mTLS ESTRICTO, servicio por servicio

4  autorización en modo REGISTRO
```

```
   se registra lo que se denegaría, sin denegar

5  autorización APLICADA, con denegación por defecto
```

Y el paso 2 suele ser el que más valor da:

```
el grafo real de llamadas casi nunca coincide con el
diagrama
→ aparecen llamadas que nadie sabía                clase 182
→ y servicios que no habla con nadie: candidatos a retirar
                                                     ley 23
```

3. Resiliencia sin empeorar la saturación

Las funciones de resiliencia de una malla son fáciles de activar y fáciles de configurar mal.

```
EL PELIGRO PRINCIPAL: REINTENTOS ANIDADOS
  el cliente reintenta 3 veces
  la malla reintenta 3 veces
  el balanceador reintenta 2 veces
  → una petición se convierte en 18 al destino
  → y en saturación, esto es lo que impide recuperarse
                                                clase 186

REGLA
  los reintentos se hacen en UNA sola capa
  → decidir cuál, escribirlo, y desactivarlo en las demás
```

Y las reglas de configuración que este programa ya ha establecido y que aquí se aplican:

```
REINTENTOS
  solo peticiones idempotentes                  clase 117
  máximo 1 o 2, con retroceso y dispersión
  con presupuesto: «como mucho el 10 % del tráfico puede
    ser reintento»            ← esto lo da la malla y es útil

PLAZOS
  propagados: el plazo restante viaja con la petición
  y cada salto no empieza si no le queda tiempo   clase 186

CORTES (interruptor)
  abre cuando la tasa de error o la latencia superan el
  umbral
  → deja de enviar y falla rápido
  → y hay que decidir qué hace el llamante cuando abre:
    valor por defecto, cacheado o error            clase 185

EXPULSIÓN DE ATÍPICOS
  retira réplicas concretas que fallan           clase 196
  → cubre el fallo gris
```

Y una capacidad que solo la malla da con facilidad:

```
INYECCIÓN DE FALLOS
  «el 5 % de las llamadas a catálogo devuelven error»
  «el 10 % tardan 3 segundos»
  → permite ejecutar las pruebas negativas de dependencia
    sin tocar el código                             ley 22
  → es la forma más barata de comprobar si una dependencia
    declarada blanda lo es                        clase 185
```

Y el espejo de tráfico, útil para migraciones:

```
ENVIAR UNA COPIA del tráfico real al servicio nuevo, sin
usar su respuesta
→ valida con carga real antes de desviar nada     clase 184
→ cuidado: si el servicio nuevo escribe, el espejo duplica
  escrituras
```

4. Lo que cuesta de verdad

El coste de una malla se subestima porque se reparte entre muchas partidas pequeñas.

```
LATENCIA
  acompañante en el origen        +0,3-1 ms
  acompañante en el destino       +0,3-1 ms
```

```
  → una llamada añade 0,6-2 ms
  → una operación que cruza 4 servicios: +2,4-8 ms
  → y con TLS mutuo, algo más en el establecimiento

RECURSOS
  un acompañante por réplica: típicamente 50-150 MB de
  memoria y una fracción de CPU
  → 400 réplicas × 100 MB = 40 GB solo de acompañantes
  → y en cargas con muchas réplicas pequeñas, el
    acompañante puede consumir más que el propio servicio

OPERACIÓN
  un plano de control que actualizar y vigilar
  certificados internos que rotar
  una capa más donde diagnosticar
  y versiones que hay que ir subiendo

DIAGNÓSTICO
  «¿el error 503 lo devuelve mi servicio, el acompañante
   del destino o el mío?»
  → hay que aprender a leer los códigos propios de la malla
```

Y los modos que reducen coste, con lo que se pierde:

```
SIN ACOMPAÑANTE, EN EL NODO
  un proxy por nodo en vez de por réplica
  + mucho menos consumo
  − aislamiento menor entre servicios del mismo nodo

SIN PROXY, EN EL PROTOCOLO
  el cifrado y la identidad se hacen en la capa de red
  + latencia casi nula
  − no da capacidades de capa 7 (reintentos por ruta,
    autorización por método)

BIBLIOTECA EN LA APLICACIÓN
  + sin salto extra
  − una por lenguaje; actualizarla exige redesplegar todo
```

Y la decisión práctica:

```
si lo que hace falta son las capacidades 1 y 2, un modo
ligero (nodo o protocolo) suele bastar y cuesta mucho menos
→ adoptar el modo completo solo si se van a usar las
  capacidades de capa 7
```

El plano de control como punto único, que hay que entender antes:

```
si el plano de control cae
  los acompañantes siguen con su última configuración
  → el tráfico NO se para
  PERO
  no se propagan cambios
  las réplicas NUEVAS pueden no obtener configuración ni
    certificado
  y si los certificados caducan antes de que vuelva, el
    tráfico SÍ se para                              ley 13

→ vigilar el plano de control y la antigüedad de la
  configuración distribuida es obligatorio
```

Y la lista de comprobación de la clase:

- ■ está escrito cuáles de las cinco capacidades faltan
- ■ las que ya están resueltas no se duplican
- ■ los reintentos se hacen en UNA sola capa
- ■ hay presupuesto de reintentos
- ■ los plazos se propagan
- ■ mTLS se implantó en permisivo antes que en estricto
- ■ el grafo real de llamadas se comparó con el diagrama
- ■ la autorización pasó por modo registro antes de aplicar
- ■ hay denegación por defecto entre servicios
- ■ está medida la latencia añadida por salto
- ■ está medido el consumo de los acompañantes
- ■ se vigila el plano de control y la antigüedad de la

```
  configuración
■ se usa inyección de fallos para las pruebas negativas
```

Y el cierre que enlaza con la clase siguiente: con identidad, política y telemetría entre servicios, sigue habiendo incidentes cuya causa no aparece en ninguna métrica. Ver el tráfico de verdad —flujos, captura y observación del núcleo— es la materia de la clase 202.

Ejemplo trabajado

CloudShop evalúa adoptar una malla de servicios. Lo que sigue es el análisis capacidad por capacidad, la decisión de adoptar solo dos, y lo que apareció al ver el grafo real de llamadas.

El análisis, hecho con el método de la clase 152.

```
1  IDENTIDAD Y mTLS
   estado actual
     el tráfico entre servicios va sin cifrar dentro de la
     red
     la autorización es por grupo de seguridad: 40 servicios
     comparten subred
     3 servicios usan una clave estática compartida para
     autenticarse entre sí, de 2022
   veredicto   FALTA, y no hay alternativa razonable a escala

2  AUTORIZACIÓN SERVICIO A SERVICIO
   estado actual
     ninguna: cualquier servicio puede llamar a cualquier
     otro
     el modelo de amenazas dio alcance de 11 servicios desde
     uno comprometido                              clase 189
   veredicto   FALTA

3  RESILIENCIA
   estado actual
     biblioteca común en los 2 lenguajes principales, con
     plazos, reintentos y cortes
     el balanceador de capa 7 ya hace reintentos de entrada
   veredicto   RESUELTA — y duplicarla es peligroso

4  GESTIÓN DE TRÁFICO
   estado actual
     despliegue escalonado por el balanceador y por la
     canalización                                  clase 102
     lo que NO hay: espejo de tráfico e inyección de fallos
   veredicto   RESUELTA en un 80 %; falta lo de pruebas

5  TELEMETRÍA
   estado actual
     trazas instrumentadas, con contexto propagado
                                                   clase 121
     lo que falta: métricas por PAR de servicios sin tocar
     código
   veredicto   RESUELTA en su mayor parte
```

La decisión:

```
se adopta una malla, EN MODO LIGERO, y solo para 1 y 2

  identidad y mTLS                          sí
  autorización servicio a servicio          sí
  reintentos, plazos y cortes de la malla   DESACTIVADOS
    → se quedan en la biblioteca, que ya funciona
  gestión de tráfico                        solo inyección
                                            de fallos, en
                                            entornos de
                                            prueba
  telemetría                                se acepta la que
                                            venga, sin
                                            sustituir la
                                            existente

modo   proxy por nodo, no acompañante por réplica
motivo 412 réplicas × 100 MB = 41 GB de memoria solo en
```

```
       acompañantes; con proxy por nodo son 38 nodos × 250 MB
       = 9,5 GB
coste  ahorro estimado 2.100 €/mes frente al modo completo
lo que se pierde   aislamiento entre servicios del mismo
       nodo → aceptado, y registrado                clase 190
```

Fase 1: mTLS permisivo y el grafo real.

```
se activó en modo permisivo durante 5 semanas
nada se rompió
y se obtuvo el grafo de llamadas observado

  servicios declarados en el diagrama                 24
  servicios que emitieron o recibieron tráfico        29

  los 5 no declarados
    · un servicio de exportación creado en 2022, sin dueño
    · dos réplicas de un servicio que se creía retirado y
      seguía recibiendo el 0,4 % del tráfico
    · un trabajo programado que llamaba a la API interna
    · un panel interno de un equipo de negocio

  llamadas declaradas en el diagrama                  61
  llamadas observadas                                 96

  las 35 no declaradas, clasificadas
    18   llamadas legítimas nunca documentadas
     9   de servicios de prueba a servicios de producción
         ← hallazgo grave
     5   llamadas circulares que nadie sabía que existían
     3   de un servicio a otro que hacía de intermediario
         sin razón

  servicios que NO hablaron con nadie en 5 semanas       3
    → candidatos a retirada                          ley 23
```

Y la línea que resume la fase:

```
el diagrama tenía 24 servicios y 61 llamadas
la realidad tenía 29 y 96
→ y las 9 llamadas de prueba a producción llevaban meses
                                              ley 24, clase 199
```

Fase 2: autorización, en modo registro y luego aplicada.

```
modo registro, 3 semanas
  llamadas que se habrían denegado con la política escrita
    a partir del grafo observado                     340/día
  de ellas, legítimas y no previstas                   6
    → política corregida
  de ellas, de entornos de prueba a producción       310/día
    → cortadas a propósito
  de ellas, del servicio sin dueño                    24/día
    → se identificó al equipo, se declaró y se documentó

aplicación, servicio por servicio, 6 semanas
  denegación por defecto
  política por origen, destino, método y ruta
```

Y el efecto medido en el modelo de amenazas:

```
alcance desde un servicio comprometido
  antes    11 servicios, cualquier método
  después   2 servicios, 3 métodos concretos

credenciales estáticas compartidas entre servicios
  antes     3
  después   0
```

El coste real, medido:

```
latencia añadida por salto (proxy por nodo)
  p50                                  +0,4 ms
  p99                                  +1,1 ms
el flujo de compra cruza 3 saltos internos
  p99 del flujo                        +3,4 ms
```

```
  sobre un objetivo de 500 ms             aceptable

recursos
  memoria total de proxies                 9,5 GB
  CPU                                     ~4 % del clúster

operación
  actualizaciones del plano de control        1/trimestre
  incidentes atribuidos a la malla en 12 meses      2
    · una versión del proxy con una regresión de plazos
    · certificados internos que no rotaron en 3 nodos
      → detectado por la alerta de antigüedad       ley 13
```

El uso que resultó más valioso y no estaba en la justificación:

```
INYECCIÓN DE FALLOS en preproducción
  se ejecutaron las pruebas negativas de dependencia sin
  tocar código                                          ley 22

  resultados de la primera tanda
    catálogo declarado blando → el flujo se caía      ✗
    recomendaciones blando → correcto                 ✓
    precios blando → correcto                         ✓
    envíos declarado blando → el listado se caía      ✗
    identidad → correcto (validación local)           ✓

  → 2 de 5 dependencias declaradas blandas eran duras
  → el cálculo del techo de disponibilidad estaba mal
                                                  clase 185
  → corregidas con plazo y alternativa; techo recalculado
```

Y la observación sobre esto:

```
la capacidad que más problemas reales destapó no fue
ninguna de las dos por las que se adoptó la malla
→ fue poder romper cosas a propósito sin tocar código
```

El resultado, a los doce meses:

```
servicios con identidad criptográfica           0          29
credenciales estáticas entre servicios          3           0
alcance desde un servicio comprometido         11           2
llamadas de prueba a producción            310/día          0
servicios no declarados                         5           0
servicios retirados por no usarse               —           3
dependencias blandas que eran duras             2           0
latencia añadida en el flujo de compra          —      +3,4 ms
coste de la malla                               —   1.900 €/mes
```

La lección que esta clase deja: de las cinco capacidades, tres ya estaban resueltas y activarlas habría producido reintentos anidados; la malla se adoptó por dos y en el modo más barato que las daba. El hallazgo mayor de la implantación no fue de seguridad: fue que el grafo real tenía 96 llamadas donde el diagrama declaraba 61, incluidas nueve de entornos de prueba a producción. Y lo que más problemas destapó a lo largo del año fue poder inyectar fallos sin tocar código, que reveló que dos dependencias declaradas blandas no lo eran.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-16-advanced-cloud-networking-edge/201-service-mesh-mtls-y-gestion-de-trafico-este-oeste/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 service-mesh 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 service-mesh 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 petición fallida genera decenas de intentos contra el destino	Reintentos anidados en cliente, malla y balanceador	Reintenta en una sola capa, escríbelo, desactiva las demás y usa presupuesto de reintentos.
La malla consume más recursos que los propios servicios	Un acompañante por réplica en cargas con muchas réplicas pequeñas	Evalúa el modo por nodo o en la capa de protocolo si solo necesitas identidad y autorización.
Activar la autorización corta tráfico legítimo	La política se escribió sobre el diagrama y no sobre el grafo real	Pasa por modo permisivo y modo registro, construye la política con el tráfico observado y aplícala servicio por servicio.
Se adopta una malla y no mejora nada medible	Las capacidades que faltaban ya estaban resueltas en otra capa	Analiza las cinco por separado, adopta solo las que faltan y registra la decisión con sus premisas.
Las réplicas nuevas no arrancan correctamente	El plano de control está caído y no entrega configuración ni certificados	Vigila el plano de control y la antigüedad de la configuración distribuida, y alerta sobre certificados internos sin rotar.
Nadie sabe qué componente devuelve un error	La malla añade una capa con sus propios códigos	Aprende y documenta los códigos propios de la malla, y correlaciona por identificador de traza en las tres capas.

■ Seguridad, ética y costo

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

■ Preguntas de comprobación

1. ¿Cuáles son las cinco capacidades de una malla y cuáles suelen estar ya resueltas?
2. ¿Por qué la autorización por grupo de seguridad no basta a escala?
3. ¿Qué peligro tiene activar los reintentos de la malla sin desactivar los demás?

4. ¿Qué ocurre si cae el plano de control y qué lo convierte en un corte real?
5. ¿Qué capacidad permite ejecutar pruebas negativas de dependencia sin tocar código?

Referencias

• Istio (2025). Security: mutual TLS and authorization policies. <https://istio.io/latest/docs/concepts/security/>
• Linkerd (2025). Architecture and performance. <https://linkerd.io/2/reference/architecture/>
• SPIFFE (2025). Workload identity specification. <https://spiffe.io/docs/latest/spiffe-about/overview/>
• Cilium (2025). Service mesh without sidecars. <https://cilium.io/use-cases/service-mesh/>
• Google (2016). SRE Book: addressing cascading failures — reintentos y presupuestos. <https://sre.google/sre-book/addressing-cascading-failures/>
• Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 16 en PDF · Manual integral

Anterior	Índice	Siguiente
← 200 · Private endpoints, service networking y egress control	Parte 16 · Programa	202 · eBPF, flow logs, packet capture y diagnóstico →

Evaluación — 201 Service mesh, mTLS y gestión de tráfico este-oeste

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 service-mesh 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.

202 — eBPF, flow logs, packet capture y diagnóstico

Parte: 16 — Redes cloud avanzadas, conectividad híbrida y edge Nivel: avanzado · Horas estimadas: 4 Laboratorio: observability · Estado: EXECUTABLECORE

Propósito

Ver el tráfico de verdad cuando las métricas no bastan, que es exactamente cuando ocurren los incidentes que duran horas. La clase ordena las herramientas de diagnóstico de red por coste y por lo que cada una puede responder —registros de flujo, observación desde el núcleo, captura de paquetes—, da el método de descarte que evita perder tiempo, y aborda lo que casi nunca se explica: qué hacer cuando el tráfico va cifrado y no se puede mirar dentro.

Resultados de aprendizaje

Al finalizar podrás:

1. Elegir la herramienta adecuada a la pregunta que se está haciendo.
2. Leer registros de flujo para responder qué habla con qué y qué se rechaza.
3. Usar observación desde el núcleo sin desplegar agentes intrusivos.
4. Capturar paquetes con filtro y de forma acotada, y saber qué buscar.
5. Diagnosticar cuando el contenido está cifrado y no se puede inspeccionar.

Conceptos centrales

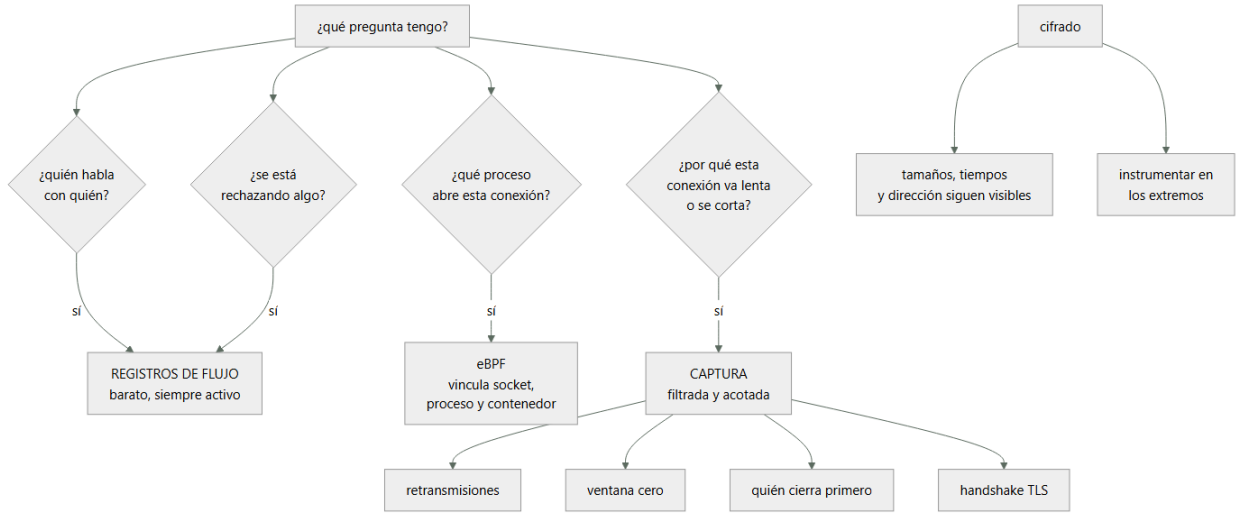
Concepto	Comprensión verificable
registro de flujo	Resumen por conexión: origen, destino, puerto, bytes, y si fue aceptada o rechazada. Barato y siempre disponible.
eBPF	Mecanismo que permite ejecutar código de observación dentro del núcleo, sin modificar aplicaciones ni añadir proxies.
captura de paquetes	Copia del tráfico real. La única que ve el detalle, y la más cara en recursos y en privacidad.
muestreo	Registrar una fracción de los flujos. Reduce coste y hace que los eventos raros no aparezcan.
retransmisión	Reenvío de un segmento perdido. Su tasa es la señal más directa de un problema de red.
ventana cero	El receptor anuncia que no puede recibir más. Indica que el problema está en el destino, no en la red.

Modelo mental

La red cloud es un sistema distribuido de rutas, identidades y políticas; cada salto añade latencia, costo y una frontera de fallo.

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 herramientas, tres preguntas

El error que más tiempo cuesta es empezar capturando paquetes. Cada herramienta responde preguntas distintas y hay que ir de la barata a la cara.

```
REGISTROS DE FLUJO
  qué dan    por conexión: origen, destino, puertos, protocolo,
             bytes, paquetes, si fue aceptada o rechazada, y
             a menudo por qué regla
  coste      bajo; se pueden dejar siempre activos
  responden
    ¿quién habla con quién?
    ¿se está rechazando algo, y qué regla lo rechaza?
    ¿cuánto tráfico va a cada destino?      ← coste  clase 200
    ¿hay tráfico hacia sitios inesperados?
  NO responden
    por qué una conexión concreta va lenta
    qué ocurre dentro de la conversación

OBSERVACIÓN DESDE EL NÚCLEO (eBPF)
  qué da     eventos en el núcleo: sockets, llamadas al
             sistema, latencia por conexión, y —lo importante—
             QUÉ PROCESO Y QUÉ CONTENEDOR
  coste      bajo o medio; sin modificar aplicaciones
  responde
    ¿qué proceso abre esta conexión?
    ¿cuánto tarda cada fase: resolución, conexión, respuesta?
    ¿qué contenedor genera este tráfico?
    ¿dónde se pierden paquetes dentro del nodo?

CAPTURA DE PAQUETES
  qué da     el tráfico real
  coste      alto: CPU, disco, y contiene datos personales
  responde
    todo lo demás, pero solo si sabes qué buscar
```

Y el método de descarte, que es el orden correcto:

```
1  ¿llega la conexión al destino?        → flujos
2  ¿la rechaza alguna regla?             → flujos
3  ¿qué proceso la abre y cuánto tarda   → eBPF
   cada fase?
4  ¿qué pasa dentro de la conversación?  → captura

→ y en la práctica, los pasos 1 y 2 resuelven la mayoría
```

Y una advertencia sobre los registros de flujo:

```
EL MUESTREO
  muchos productos muestrean: 1 de cada N flujos
```

```
  → los eventos raros desaparecen
  → y precisamente lo que se busca en un incidente suele
    ser raro

→ conviene tener muestreo completo al menos en las subredes
  críticas, aunque cueste más
```

2. Leer registros de flujo

Los registros de flujo están casi siempre activados y casi nunca se usan bien.

```
LO QUE SE PUEDE RESPONDER EN MINUTOS

¿este servicio habla con quien creemos?
  agrupar por par origen-destino y contar
  → y comparar con el diagrama                        ley 24

¿qué se está rechazando?
  filtrar por acción de rechazo y agrupar por regla
  → los rechazos son la señal más informativa y la que
    nadie mira                                        ley 15

¿a dónde va el tráfico saliente y cuánto?
  agrupar por destino externo y sumar bytes
  → esto es lo que descubre coste y exfiltración  clase 200

¿hay tráfico entre zonas que podría ser dentro de una?
  agrupar por zona de origen y de destino
  → suele ser la línea de coste más grande y peor
    atribuida                                    clase 168

¿qué habla con este recurso antes de retirarlo?
  filtrar por su dirección durante semanas
  → es la forma correcta de decidir si algo se puede
    apagar                                         ley 23
```

Y los patrones que se reconocen en los flujos:

```
MUCHOS FLUJOS CORTOS AL MISMO DESTINO
  sin reutilización de conexión
  → cada petición abre una conexión nueva
  → coste de establecimiento y de TLS en cada una

FLUJOS RECHAZADOS REPETIDOS DESDE EL MISMO ORIGEN
  un cliente que no sabe que ya no puede
  → suele ser un servicio retirado a medias

TRÁFICO SALIENTE CONSTANTE Y PEQUEÑO A UN DESTINO ÚNICO
  la firma clásica de un agente o de una exfiltración lenta
                                                 clase 200

UN ORIGEN QUE HABLA CON MUCHOS DESTINOS INTERNOS
  reconocimiento, o un servicio que hace de intermediario
  sin razón                                      clase 201
```

Y una limitación que hay que tener presente:

```
los flujos dicen QUE hubo conexión, no si funcionó
  una conexión aceptada que devuelve errores de aplicación
  aparece igual que una correcta
→ para eso hacen falta las métricas de aplicación
                                                 clase 121
```

3. Captura de paquetes: qué buscar

Capturar sin saber qué se busca produce gigabytes inútiles. Con cuatro cosas se resuelve casi todo.

```
ANTES DE CAPTURAR
  filtra en el origen: host, puerto y protocolo
  acota el tiempo o el tamaño
  captura en los DOS extremos si es posible    ← clave
  y ten en cuenta que el contenido puede tener datos
    personales: la captura es un dato sensible  clase 189
```

Las cuatro señales que hay que mirar:

```
1  RETRANSMISIONES
   segmentos reenviados porque no llegó confirmación
   tasa normal        < 0,1 %
   preocupante        > 1 %
   → indica pérdida en el camino: enlace saturado, MTU,
     equipo intermedio                          clase 198

2  VENTANA CERO
   el receptor anuncia que no puede recibir más
   → el problema está en el DESTINO, no en la red
   → el destino no consume lo que le llega: aplicación
     bloqueada, disco lento, hilos agotados     clase 186

3  QUIÉN CIERRA PRIMERO Y CÓMO
   cierre limpio     alguien terminó a propósito
   reinicio abrupto  algo cortó: cortafuegos, plazo, proceso
                     caído
   → y saber CUÁL de los dos extremos lo envió señala dónde
     está la causa
   → un reinicio desde un intermedio suele ser un cortafuegos
     con estado que perdió la sesión            clase 194

4  EL ESTABLECIMIENTO
   ¿cuánto tarda el saludo inicial?  → red
   ¿cuánto tarda el saludo TLS?      → certificados, cadena
   ¿se completa?                     → versión o cifrado
                                       incompatible
   → aquí se ven los fallos de certificado con precisión
                                              clase 196
```

Y los patrones que identifican causa directamente:

```
SÍNTOMA EN LA CAPTURA          CAUSA
retransmisiones solo de        MTU sin ajustar    clase 198
paquetes grandes

ventana cero repetida          el destino no consume:
                               mirar el destino, no la red

reinicio abrupto tras 60 s     un intermedio con plazo de
exactos de inactividad         inactividad menor que los
                               extremos           clase 196

saludo TLS que empieza y no    cadena incompleta o versión
termina                        no soportada

muchos establecimientos por    sin reutilización de conexión
segundo al mismo destino

todo correcto en la captura    el problema es de aplicación,
                               no de red      ← también es
                               un resultado
```

Y el último merece énfasis:

```
una captura limpia DESCARTA la red
→ y descartar es la mitad del diagnóstico
→ decir «la red está bien, con esta evidencia» vale tanto
  como encontrar la causa
```

4. Cuando el tráfico va cifrado

Con TLS en todas partes —y más aún con TLS mutuo— la captura ya no muestra el contenido. Eso cambia el método pero no lo impide.

```
LO QUE SIGUE SIENDO VISIBLE CIFRADO
  quién habla con quién, y cuándo
  el nombre solicitado en el saludo, salvo que esté cifrado
  tamaños de los mensajes y su patrón temporal
  el establecimiento completo y sus errores
  retransmisiones, ventana cero y cierres

→ es decir: casi todo lo que hace falta para diagnosticar
  problemas de RED
```

Y lo que hay que hacer para lo demás:

```
INSTRUMENTAR EN LOS EXTREMOS
  trazas con contexto propagado                  clase 121
  → dan lo que la captura ya no puede dar

OBSERVAR DESDE EL NÚCLEO
  eBPF puede ver los datos ANTES de cifrar y DESPUÉS de
  descifrar, dentro del proceso
  → sin claves, sin proxy y sin tocar la aplicación
  → es la razón principal por la que esta técnica se ha
    vuelto central

DESCIFRAR EN UN PUNTO CONTROLADO
  el balanceador o el proxy ya ven el tráfico en claro
  → registrar allí, con cuidado de no registrar datos
    personales                                   clase 189

Y LO QUE NO SE DEBE HACER
  guardar claves privadas para poder descifrar capturas
  → convierte el archivo de capturas en el activo más
    peligroso de la organización
```

La operación del diagnóstico, que evita el caos durante los incidentes:

```
QUÉ HAY QUE TENER PREPARADO ANTES
  flujos activos y consultables, sin muestreo en lo crítico
  permiso y procedimiento para capturar, con quién autoriza
  un sitio donde guardar capturas, con caducidad automática
  las herramientas instaladas o disponibles al momento
  y el camino esperado de cada flujo, escrito     clase 194

→ conseguir un permiso de captura durante un incidente
  añade horas
```

Y el tratamiento de las capturas como dato sensible:

```
contienen datos personales y a veces credenciales
→ acceso restringido, caducidad corta, y registro de quién
  las descarga
→ y nunca en un almacén compartido general        clase 189
```

Y la lista de comprobación de la clase:

- ■ los registros de flujo están activos y son consultables
- ■ no hay muestreo en las subredes críticas
- ■ los rechazos se revisan periódicamente, no solo en incidentes
- ■ hay observación desde el núcleo capaz de vincular conexión, proceso y contenedor
- ■ existe procedimiento y permiso previo para capturar
- ■ las capturas se filtran y se acotan en origen
- ■ las capturas caducan solas y su acceso queda registrado
- ■ no se guardan claves privadas para descifrar capturas
- ■ el diagnóstico de aplicación se apoya en trazas, no en contenido de paquetes
- ■ el camino esperado de los flujos principales está escrito

Y el cierre que enlaza con la clase siguiente: hasta aquí la red ha sido de centros de datos y nubes, con enlaces estables. Cuando el extremo es una tienda, un vehículo o un dispositivo que se queda sin cobertura, las reglas cambian. Redes definidas por software, 5G y operación desconectada es la materia de la clase 203.

Ejemplo trabajado

Tres incidentes de CloudShop que las métricas no explicaban. Lo que sigue es el diagnóstico de cada uno con la herramienta que correspondía, y el tiempo que costó cada vez que se empezó por la equivocada.

Incidente 1 · «El servicio de envíos falla al llamar al transportista», abril.

```
síntoma   el 12 % de las llamadas al API del transportista
          fallaban por plazo vencido
          las métricas de aplicación solo decían «timeout»

lo que se hizo primero (mal)
```

```
  2 h revisando el código del cliente HTTP
  1 h capturando paquetes sin filtro en el nodo
     → 3,2 GB de captura, imposible de leer

lo que lo resolvió, en 6 minutos
  REGISTROS DE FLUJO, filtrados por el destino del
  transportista

    flujos aceptados                          88 %
    flujos RECHAZADOS                         12 %
    regla que los rechazaba   el grupo de seguridad de
                              salida de una de las tres
                              subredes de la aplicación

  causa    al añadir una tercera subred en febrero, se copió
           la configuración de otra y faltaba la regla de
           salida hacia el rango del transportista
  → el 12 % era exactamente 1 de cada 3 réplicas

lección   la primera pregunta era «¿se rechaza algo?» y la
          responden los flujos en un minuto
```

Incidente 2 · «Conexiones que se cuelgan cada 60 segundos exactos», julio.

```
síntoma   las conexiones de larga duración entre el servicio
          de informes y la base corporativa se cortaban con
          una regularidad exacta
          reintentaban y funcionaban, pero cada corte perdía
          la consulta en curso

flujos    mostraban conexiones establecidas y cerradas, sin
          rechazos → no bastaba

eBPF      mostró que el cierre venía de fuera del proceso
          y que el proceso no lo pedía

CAPTURA   filtrada por el par de direcciones y el puerto,
          durante 5 minutos: 4 MB

  lo que se vio
    tras exactamente 60 s sin datos, llegaba un reinicio
    abrupto
    el reinicio NO venía de ninguno de los dos extremos:
    los números de secuencia no correspondían
    → lo enviaba un intermedio

  causa   el cortafuegos del enlace dedicado tenía un plazo
          de inactividad de 60 s
          los dos extremos tenían 300 s y no enviaban nada
          entre consultas

  corrección
    mensajes de mantenimiento de conexión cada 30 s en el
    cliente
    y el plazo del cortafuegos documentado en el registro de
    caminos esperados                          clase 194

tiempo total    40 min, de los cuales 25 fueron conseguir
                permiso para capturar
→ desde entonces, el permiso está preautorizado con
  procedimiento
```

Incidente 3 · «Latencia alta e intermitente solo en una zona», octubre.

```
síntoma   el p99 del servicio de catálogo en la zona C era
          4 veces el de las zonas A y B
          CPU, memoria y errores, normales en las tres

flujos    volumen y destinos idénticos en las tres zonas
          → no explicaba nada

eBPF      desglose de latencia por fase, por conexión

    fase                    zona A    zona B    zona C
```

```
    resolución de nombre     0,4 ms    0,4 ms   38 ms   ←
    establecimiento          0,6 ms    0,6 ms    0,7 ms
    saludo TLS               1,2 ms    1,2 ms    1,3 ms
    primera respuesta        41 ms     43 ms     44 ms

  → el problema era la RESOLUCIÓN DE NOMBRES, no el tráfico

  causa
    el resolutor de la zona C tenía una regla de reenvío
    apuntando a un servidor corporativo que respondía lento
    las otras dos usaban el resolutor local        clase 195
    y como la biblioteca no cacheaba, cada petición
    resolvía de nuevo

  corrección
    regla de reenvío corregida
    caché de resolución activada en el cliente
    p99 de la zona C                    de 84 ms a 46 ms

lección   sin el desglose por fase, esto se habría
          diagnosticado como «la red de la zona C va mal»
          y se habría escalado al proveedor
```

Lo que se montó después de los tres.

```
FLUJOS
  activados en todas las subredes
  sin muestreo en producción y en datos
  con 30 días de retención y consulta indexada
  coste                                  +410 €/mes

  y un panel que nadie tenía: RECHAZOS POR REGLA
    revisión semanal
    en los 6 primeros meses encontró
      · 3 servicios retirados cuyos clientes seguían
        intentando conectar
      · 1 regla que rechazaba tráfico legítimo desde una
        subred nueva (el incidente 1, en otras dos subredes)
      · 2 orígenes internos intentando alcanzar producción
        desde preproducción                    clase 199

OBSERVACIÓN DESDE EL NÚCLEO
  desplegada en todos los nodos
  da: latencia por fase, proceso y contenedor por conexión,
      pérdidas dentro del nodo
  coste                          ~2 % de CPU, 180 €/mes

CAPTURA
  procedimiento preautorizado, con dos personas que pueden
  activarla
  filtro obligatorio en origen
  almacenamiento con caducidad de 7 días y acceso registrado
  prohibido guardar claves privadas para descifrar
```

El efecto medido:

```
tiempo medio de diagnóstico de red        1 h 55       12 min
incidentes en que se capturó sin filtro        3           0
tiempo perdido en conseguir permisos       25 min          0
rechazos revisados periódicamente             no       semanal
problemas encontrados por la revisión
  de rechazos, sin incidente previo            —           6
```

Y una observación sobre el último dato:

```
seis problemas se encontraron mirando los rechazos, sin que
hubiera ningún incidente
→ tres de ellos habrían causado uno más adelante
→ la señal existía desde siempre y nadie la miraba  ley 15
```

La lección que esta clase deja: los tres incidentes se resolvieron con tres herramientas distintas, y en dos de ellos se empezó por la más cara: tres horas de código y captura ciega para un problema que los registros de flujo respondían en seis minutos. El incidente que parecía más de red —latencia alta en una zona— no era de red: era resolución de nombres, y solo se vio con el desglose de latencia por fase. Y el panel más útil del año no lo pidió ningún incidente: fue

mirar lo que se estaba rechazando.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-16-advanced-cloud-networking-edge/202-ebpf-flow-logs-packet-capture-y-diagnostico/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 network-evidence 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 network-evidence 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 pierden horas en un diagnóstico que los flujos resolvían en minutos	Se empezó capturando paquetes en vez de descartar por orden	Pregunta primero si la conexión llega y si algo la rechaza; los flujos responden eso al instante.
Un evento raro no aparece en los registros de flujo	El producto está muestreando una fracción de los flujos	Desactiva el muestreo al menos en las subredes críticas, aunque cueste más.
La captura ocupa gigabytes y no se puede analizar	Se capturó sin filtro ni límite de tiempo	Filtra por host, puerto y protocolo en el origen, acota la duración y captura en ambos extremos.
Las conexiones se cortan con periodicidad exacta	Un intermedio con plazo de inactividad menor que el de los extremos envía un reinicio	Comprueba en la captura de quién viene el reinicio y añade mensajes de mantenimiento de conexión por debajo de ese plazo.
Se atribuye a la red una latencia que no es de red	No se desglosó la latencia por fase de la conexión	Usa observación desde el núcleo para separar resolución, establecimiento, saludo TLS y primera respuesta.
El archivo de capturas se convierte en un riesgo	Se guardan capturas sin caducidad, o claves privadas para descifrarlas	Caducidad corta, acceso restringido y registrado, y nunca almacenar claves privadas para descifrar tráfico capturado.

Seguridad, ética y costo

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

Preguntas de comprobación

1. ¿Qué pregunta responde cada una de las tres herramientas y en qué orden se usan?
2. ¿Por qué el muestreo de flujos estorba precisamente en un incidente?
3. ¿Qué indica una ventana cero repetida y hacia dónde dirige el diagnóstico?
4. ¿Qué sigue siendo visible en una captura de tráfico cifrado?
5. ¿Qué hay que tener preparado antes de necesitar una captura?

Referencias

- AWS (2025). VPC Flow Logs — campos, acciones y limitaciones. <https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html>
- Cilium (2025). Hubble: eBPF-based network observability. <https://docs.cilium.io/en/stable/overview/intro/>
- Gregg, B. (2019). BPF Performance Tools. <https://www.brendangregg.com/bpf-performance-tools-book.html>
- Wireshark (2025). TCP analysis flags: retransmissions, zero window, resets. <https://www.wireshark.org/docs/wsughtmlchunked/ChAdvTCPAnalysis.html>
- Google Cloud (2025). Packet Mirroring and network diagnostics. <https://cloud.google.com/vpc/docs/packet-mirroring>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 16 en PDF · Manual integral

Anterior	Índice	Siguiente
← 201 · Service mesh, mTLS y gestión de tráfico este-oeste	Parte 16 · Programa	203 · SD-WAN, 5G, IoT y operación desconectada →

Evaluación — 202 eBPF, flow logs, packet capture y diagnóstico

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 network-evidence 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.

203 — SD-WAN, 5G, IoT y operación desconectada

Parte: 16 — Redes cloud avanzadas, conectividad híbrida y edge Nivel: avanzado · Horas estimadas: 4 Laboratorio: architecture · Estado: EXECUTABLECORE

Propósito

Diseñar para el extremo, donde el enlace no es fiable, la latencia no es despreciable y el dispositivo puede pasar horas sin conexión. La clase cubre las redes definidas por software para conectar cientos de sedes, lo que 5G aporta de verdad y lo que se le atribuye sin fundamento, y —lo que más decisiones cambia— cómo se diseña un sistema que tiene que seguir funcionando sin red y reconciliar después sin perder datos ni duplicarlos.

Resultados de aprendizaje

Al finalizar podrás:

1. Conectar muchas sedes con redes definidas por software y políticas centrales.
2. Distinguir lo que 5G aporta de lo que se le atribuye.
3. Diseñar operación desconectada con estado local y reconciliación.
4. Resolver conflictos de reconciliación sin perder ni duplicar.
5. Operar miles de extremos con actualizaciones seguras y vuelta atrás.

Conceptos centrales

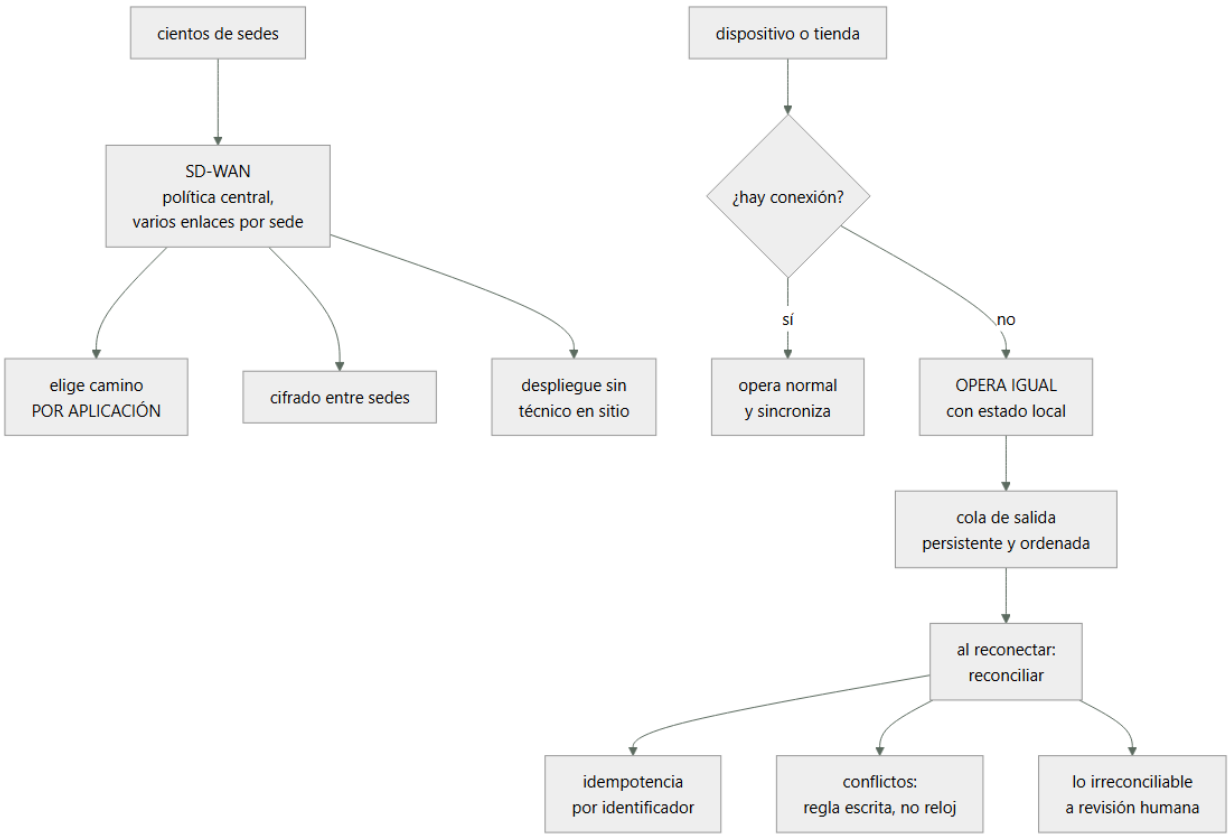
Concepto	Comprensión verificable
red definida por software (SD-WAN)	Capa que gestiona muchas sedes con política central y varios enlaces por sede, eligiendo camino por aplicación.
cómputo en el extremo	Procesamiento cerca del dispositivo, para no depender del enlace ni de la latencia hacia la nube.
operación desconectada	Capacidad de seguir prestando servicio sin conexión, con estado local y reconciliación posterior.
reconciliación	Fusión del estado local con el central al recuperar conexión. Es donde se pierden o duplican datos.
cola de salida local	Registro persistente de lo ocurrido sin conexión, que se envía en orden al reconectar.
actualización segura	Despliegue a dispositivos con verificación, doble partición y vuelta atrás automática.

Modelo mental

La red cloud es un sistema distribuido de rutas, identidades y políticas; cada salto añade latencia, costo y una frontera de fallo.

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. Muchas sedes: redes definidas por software

Conectar tres oficinas se hace con túneles. Conectar cuatrocientas tiendas, no.

```
EL PROBLEMA A ESCALA
  configurar cada sede a mano no escala
  cada sede tiene 2 o 3 enlaces de calidad distinta
    fibra, radio móvil, satélite
  y el tráfico ya no va al centro de datos: va a la nube y
    a servicios de terceros
  → el modelo de «todo al centro y de ahí a internet»
    añade latencia y coste sin dar nada

LO QUE APORTA UNA SD-WAN
  política CENTRAL aplicada a todas las sedes
  varios enlaces por sede, usados a la vez
  elección de camino POR APLICACIÓN
    · el punto de venta va por el enlace fiable
    · las actualizaciones van por el barato
    · el vídeo va por el de menos jitter
  cifrado entre sedes sin configurar túneles uno a uno
  y despliegue sin técnico en sitio: el equipo se autoconfigura
```

Y las decisiones que hay que tomar y suelen olvidarse:

```
¿QUÉ PASA SI EL PLANO DE CONTROL NO RESPONDE?
  las sedes deben seguir funcionando con su última política
  → si no, el plano de control es un punto único que tira
    cuatrocientas tiendas                        clase 201

¿SALIDA LOCAL A INTERNET O CENTRALIZADA?
  local     menos latencia y coste
  central   inspección y control de salida        clase 200
  → lo habitual: salida local para servicios conocidos y
    central para el resto

¿QUÉ SE HACE CUANDO UN ENLACE SE DEGRADA SIN CAERSE?
```

```
  medir latencia, pérdida y jitter continuamente y cambiar
  de camino por umbral
  → es el fallo gris aplicado a los enlaces      clase 185
```

Y una advertencia sobre la elección de camino:

```
cambiar de camino a mitad de una sesión rompe lo que
tenga estado
→ el cambio debe aplicarse a sesiones nuevas, salvo que
  el producto mantenga la sesión
→ y el cambio de camino cambia la dirección de origen, con
  lo que eso implica para listas de terceros    clase 193
```

2. 5G: lo que aporta y lo que se le atribuye

Conviene separar con cuidado, porque hay mucha atribución sin fundamento.

```
LO QUE APORTA DE VERDAD
  ancho de banda alto sin obra civil
  → conectar una sede nueva en días, no en meses
  densidad de dispositivos por celda muy superior
  → útil en fábricas, almacenes y logística
  latencia menor que las generaciones anteriores
  segmentación de red por contrato: un canal con garantías
  distintas para tráfico crítico
  y redes privadas en instalaciones propias

LO QUE NO APORTA POR SÍ SOLO
  «latencia de un milisegundo» de extremo a extremo
    → la baja latencia es del tramo radio; el resto del
      camino sigue existiendo
    → si el servidor está a 600 km, la física manda
  fiabilidad garantizada sin contrato específico
  cobertura uniforme
  y desde luego no elimina la necesidad de operar
    desconectado
```

Y el criterio práctico:

```
5G es una buena PRIMERA opción de conectividad para sedes
nuevas y una excelente SEGUNDA para redundancia
→ y su valor real suele estar en el plazo de despliegue,
  no en la latencia

y para latencia baja de verdad hace falta acercar el CÓMPUTO,
no solo mejorar el enlace                       clase 197
```

Cómputo en el extremo, que es lo que resuelve latencia y desconexión a la vez:

```
CUÁNDO HACE FALTA
  la decisión debe tomarse en decenas de milisegundos
    (control industrial, seguridad, visión)
  el volumen de datos es tal que enviarlo todo no compensa
    (vídeo, sensores a alta frecuencia)
  el servicio DEBE seguir funcionando sin conexión
    (punto de venta, control de acceso)

CUÁNDO NO
  cuando la latencia de ir a la nube es aceptable
  → y entonces se evita operar una flota de dispositivos
```

Y el coste que introduce, que es el que decide:

```
cada extremo es un sistema que hay que
  desplegar, actualizar, vigilar, asegurar y retirar
→ con cuatrocientos extremos, cualquier tarea manual es
  imposible                                      ley 23
```

3. Operar sin conexión

Esta es la parte que más cambia el diseño y la que más se improvisa.

```
LA PREGUNTA QUE HAY QUE CONTESTAR PRIMERO
  ¿qué operaciones DEBEN funcionar sin conexión?

  ejemplo de una tienda
```

```
  cobrar                      SÍ, obligatoriamente
  consultar stock local       SÍ
  consultar stock de otra     no; se degrada
    tienda
  aplicar promoción           SÍ, con las reglas cacheadas
  devolver                    SÍ, con límite de importe
  consultar historial de un   no
    cliente
```

Y de esa lista sale todo lo demás:

```
LO QUE DEBE FUNCIONAR SIN CONEXIÓN NECESITA
  1  los DATOS que necesita, cacheados localmente y con
     su antigüedad conocida
  2  las REGLAS que aplica, cacheadas y versionadas
  3  un registro PERSISTENTE de lo ocurrido, ordenado
  4  una política de LÍMITES para lo que no se puede
     verificar
  5  y una forma de RECONCILIAR al volver
```

Y el punto 4 es el que evita el fraude y las pérdidas:

```
sin conexión no se puede verificar
  el saldo real de una tarjeta
  si un cupón ya se usó
  si un artículo está reservado

→ hay que decidir el límite de riesgo aceptable
  «devoluciones hasta 50 € sin conexión; por encima, no»
  «pagos sin conexión hasta 3 operaciones por tarjeta»
→ y esa decisión es de negocio, no técnica
```

La cola de salida local, que es el mecanismo central:

```
cada operación local se escribe en un registro persistente
  con identificador único generado localmente
  con marca de tiempo local Y contador monótono
  y sobrevive a un reinicio o a un corte de luz

al reconectar se envía EN ORDEN, con reintentos
y el servidor la procesa de forma IDEMPOTENTE  clase 117
  → el mismo identificador dos veces no crea dos cosas
```

Y los errores clásicos:

```
GUARDAR EN MEMORIA
  un corte de luz pierde las ventas de la tarde

SIN IDENTIFICADOR LOCAL ÚNICO
  el reintento crea duplicados

SIN ORDEN
  se procesa una devolución antes que la venta

COLA SIN LÍMITE NI VIGILANCIA
  el dispositivo se queda sin disco tras dos días
  → hay que alertar por antigüedad y tamaño de la cola
                                                  ley 13
```

4. Reconciliar, y operar la flota

La reconciliación es donde se pierden o duplican datos, y necesita reglas escritas antes.

```
LOS CONFLICTOS QUE APARECEN
  el mismo artículo vendido en la tienda y reservado en la
    web mientras no había conexión
  un precio que cambió centralmente y la tienda aplicó el
    antiguo
  un cupón usado dos veces en dos tiendas desconectadas
  un cliente que cambió su dirección en dos sitios

Y LA REGLA QUE NO SIRVE
  «gana la marca de tiempo más reciente»
  → los relojes de los dispositivos derivan mucho más que
    los de un servidor                          clase 187
```

Y las reglas que sí funcionan, por tipo de dato:

```
HECHOS INMUTABLES (una venta ocurrió)
  no hay conflicto: se acepta y se registra
  → por eso conviene modelar el extremo como generador de
    HECHOS, no de estado                          clase 188

CONTADORES (stock)
  se aplican las variaciones, no el valor final
  «−1 unidad» se puede aplicar en cualquier orden
  «stock = 4» no                                  ley 21

ESTADO EDITABLE (datos de cliente)
  regla explícita: gana el central, gana el local, o se
  marca para revisión

LO IRRECONCILIABLE
  a una cola de revisión humana, con contexto
  → y hay que medir cuántos casos llegan ahí; si crecen,
    la regla está mal
```

Y una consecuencia de diseño:

```
el precio y la promoción aplicados se GUARDAN EN LA VENTA
→ así la venta es un hecho completo y no depende de
  reconstruir qué precio había                    clase 149
```

Operar la flota, que es lo que hace viable todo lo anterior:

```
ACTUALIZACIONES
  firmadas y verificadas por el dispositivo       clase 106
  doble partición: se instala en la inactiva y se arranca
  vuelta atrás AUTOMÁTICA si no confirma tras arrancar
  → sin esto, una actualización mala exige visita física
  despliegue escalonado por grupos, nunca a todos  clase 102

VIGILANCIA
  cada dispositivo reporta: versión, antigüedad de datos,
  tamaño de cola, tiempo desconectado
  ALERTA POR ANTIGÜEDAD: «este dispositivo lleva N horas
  sin reportar»                                     ley 13
  → es la única forma de saber que uno murió

IDENTIDAD Y SECRETOS
  identidad por dispositivo, no compartida        clase 159
  credenciales de corta duración y renovables
  y capacidad de REVOCAR uno concreto
  → un dispositivo robado es un punto de entrada

RETIRADA
  procedimiento para dar de baja y borrar
  → sin él, la flota crece y nunca decrece          ley 23
```

Y la lista de comprobación de la clase:

- ■ está escrito qué operaciones deben funcionar sin conexión
- ■ los datos y reglas necesarios están cacheados y con antigüedad conocida
- ■ hay límites de riesgo escritos para lo no verificable
- ■ la cola local es persistente, ordenada y con identificador único
- ■ el servidor procesa de forma idempotente
- ■ hay alerta por antigüedad y tamaño de la cola
- ■ las reglas de reconciliación están escritas por tipo de dato
- ■ no se resuelve ningún conflicto por marca de tiempo del dispositivo
- ■ el stock se sincroniza por variaciones, no por valor final
- ■ hay cola de revisión humana y se mide cuánto llega
- ■ las actualizaciones son firmadas, con doble partición y vuelta atrás automática
- ■ hay alerta por dispositivo que deja de reportar
- ■ cada dispositivo tiene identidad propia y revocable
- ■ existe procedimiento de retirada

Y el cierre que enlaza con la clase siguiente: con direccionamiento, encaminamiento, nombres, entrada, borde, conectividad, salida, tráfico interno, diagnóstico y extremo, queda montarlo todo junto en un diseño de red multirregión y multinube. Es la materia de la clase 204, que además cierra la parte 16.

Ejemplo trabajado

CloudShop tiene 340 tiendas con punto de venta propio. Lo que sigue es el rediseño de conectividad, la decisión sobre qué debe funcionar sin conexión, y el incidente de reconciliación que costó 41.000 € antes de arreglarlo.

El punto de partida:

```
340 tiendas, cada una con
  1 enlace de fibra del proveedor local
  1 respaldo por radio móvil, configurado a mano
  todo el tráfico va al centro de datos y de ahí a la nube

problemas medidos
  latencia del punto de venta a la nube        180-240 ms
    · 40 ms tienda → centro de datos
    · 60 ms de cola en el enlace central saturado
    · 80 ms centro de datos → nube
  cortes de fibra al mes, en el conjunto             26
  tiempo medio de reparación                      6 h 20
  conmutación al respaldo                         MANUAL
    → una llamada del encargado y 40 min de media
  tiendas que en un corte NO podían cobrar         100 %
```

Y el dato que decidió el proyecto:

```
cortes al mes                                       26
horas de tienda sin poder cobrar                   164/mes
venta perdida estimada                          38.000 €/mes
```

El rediseño de conectividad.

```
SD-WAN con dos enlaces activos por tienda
  fibra local + radio móvil 5G
  elección de camino POR APLICACIÓN
    punto de venta        → fibra; conmuta a 5G por umbral
    actualizaciones       → el enlace más barato, siempre
    vídeo de seguridad    → 5G, para no ocupar la fibra
    navegación de empleados → salida local a internet

  conmutación automática por umbral de pérdida y latencia
    umbral   pérdida > 2 % o latencia > 150 ms durante 10 s
    tiempo de conmutación medido            2,4 s

  salida local para servicios conocidos, central para el
  resto                                          clase 200

  y la decisión que no se olvidó
    si el plano de control de la SD-WAN no responde, cada
    tienda mantiene su última política
    → probado desconectándolo 4 horas                ley 22
```

Y el efecto:

```
latencia del punto de venta a la nube    180-240 ms    70 ms
  (salida local, sin pasar por el centro de datos)
conmutación al respaldo                    40 min       2,4 s
horas de tienda sin conectividad          164/mes      9/mes
coste de conectividad                    41.000 €    36.500 €
```

Y una nota sobre el 5G:

```
lo que aportó no fue la latencia, que en el tramo relevante
apenas cambió
fue que 47 tiendas nuevas se conectaron en 4 días en vez de
en 3 meses, y que el respaldo dejó de depender del mismo
proveedor de última milla que la fibra          clase 198
```

Lo que debe funcionar sin conexión, decidido con negocio:

```
operación                          sin conexión   límite
cobrar con tarjeta                     SÍ         3 operaciones
```

```
                                                por tarjeta,
                                                máx. 300 €
cobrar en efectivo                      SÍ      sin límite
consultar stock de esta tienda          SÍ      antigüedad
                                                mostrada
consultar stock de otras tiendas        NO      se degrada
aplicar promoción                       SÍ      reglas
                                                cacheadas,
                                                versionadas
devolución con ticket                   SÍ      máx. 50 €
devolución sin ticket                   NO
emitir tarjeta regalo                   NO      ← ver abajo
consultar historial de cliente          NO
```

Y la última fila tiene historia.

El incidente de reconciliación, marzo.

```
situación   corte de fibra de 5 horas en 11 tiendas
            el punto de venta siguió funcionando

lo que pasó
  las tarjetas regalo SÍ se podían emitir y canjear sin
  conexión, porque nadie lo había excluido
  el saldo se comprobaba contra la copia local

  un cliente canjeó la misma tarjeta de 300 € en 4 tiendas
  desconectadas el mismo día
  y en las semanas siguientes, el patrón se repitió

  al reconciliar, el sistema central aplicó los 4 canjes
  → saldo negativo, y ninguna forma de recuperar el dinero

pérdida acumulada hasta detectarlo            41.000 €
tiempo hasta detectarlo                       7 semanas
cómo se detectó   un cuadre contable mensual, no una alerta
                                                    ley 15
```

Y el análisis de la causa:

```
la lista de «qué funciona sin conexión» no se había escrito
→ funcionaba CUANTO técnicamente podía funcionar
→ y nadie había decidido el límite de riesgo

y la regla de reconciliación tampoco existía
→ el central aplicaba lo que llegaba, en orden de llegada
```

Y las correcciones:

```
1  lista escrita de operaciones permitidas sin conexión,
   con límites, aprobada por negocio y por finanzas

2  tarjeta regalo: canje solo con conexión
   → si no hay conexión, se ofrece otro medio de pago

3  reglas de reconciliación por tipo de dato
     ventas          hechos: se aceptan todas
     stock           variaciones, no valor final
     saldos          verificación obligatoria en central
     devoluciones    hasta 50 €, aceptadas; el resto a
                     revisión

4  cola de revisión humana, con panel
   casos que llegan al mes                        18
   → estable; si superara 60, la regla estaría mal

5  alerta por patrón: «la misma tarjeta o el mismo cupón
   aparece en más de una tienda en la misma ventana de
   desconexión»
   → habría detectado el problema en 2 días, no en 7 semanas
```

La cola de salida local, rediseñada:

```
antes   en memoria, con envío inmediato y sin identificador
        propio
```

```
        → un corte de luz en enero perdió 340 ventas

después
  registro persistente en disco, con sincronización forzada
  identificador único por operación, generado localmente
  contador monótono por dispositivo, además de la hora
  envío en orden, con reintentos y retroceso
  procesamiento idempotente en el central       clase 117
  alerta si la cola supera 500 elementos o 4 horas de
    antigüedad                                    ley 13

prueba negativa
  cortar la luz a mitad de una venta, 20 veces
  → 20 de 20 recuperadas al arrancar
```

La operación de la flota:

```
actualizaciones
  firmadas y verificadas por el dispositivo
  doble partición con vuelta atrás automática si no
    confirma en 10 min
  despliegue por grupos: 5 tiendas → 40 → 340
  → en junio, una versión con un fallo de impresión de
    tickets se detectó en el grupo de 5 y no llegó a las
    otras 335

vigilancia por dispositivo
  versión, antigüedad de datos, tamaño de cola, horas
  desconectado
  alerta «lleva más de 2 h sin reportar»
  → detectó 3 puntos de venta muertos que las tiendas no
    habían reportado porque usaban el de al lado

identidad
  certificado por dispositivo, rotación automática
  revocación individual probada
  → en septiembre se robó un punto de venta; revocado en
    11 minutos

retirada
  procedimiento de baja con borrado
  dispositivos dados de baja en el año               62
  → antes, los antiguos se quedaban en el inventario y con
    credenciales válidas                           ley 23
```

El resultado del año:

```
horas de tienda sin poder cobrar          164/mes      0/mes
latencia del punto de venta            180-240 ms      70 ms
conmutación al respaldo                    40 min       2,4 s
ventas perdidas por corte de luz         340 (ene)          0
pérdidas por canje duplicado           41.000 € (7 sem)     0
tiempo de conexión de una tienda nueva    3 meses       4 días
dispositivos muertos sin detectar                3           0
coste de conectividad                   41.000 €    36.500 €
```

La lección que esta clase deja: el rediseño de red resolvió lo que se había planteado —de 164 horas al mes sin poder cobrar a cero— y aun así el problema más caro del año no fue de conectividad: fue que nadie había escrito qué debía poder hacerse sin conexión, así que se podía hacer todo, incluido canjear la misma tarjeta regalo en cuatro tiendas. Y tardó siete semanas en detectarse porque la señal era un cuadre contable mensual y no una alerta.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-16-advanced-cloud-networking-edge/203-sd-wan-5g-iot-y-operacion-desconectada/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 edge-topology 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 edge-topology 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 corte del plano de control deja sin servicio a cientos de sedes	Los equipos de sede no funcionan con su última política conocida	Exige operación autónoma con la política cacheada y pruébalo desconectando el plano de control durante horas.
Se espera latencia de milisegundos por usar 5G y no se consigue	La mejora es del tramo radio; el resto del camino hasta el servidor sigue igual	Si hace falta latencia baja de verdad, acerca el cómputo; valora 5G por plazo de despliegue y por independencia de la última milla.
Un corte de luz pierde las operaciones de horas	La cola de salida vive en memoria	Registro persistente con sincronización forzada, identificador único local y envío ordenado con reintentos.
Al reconectar aparecen operaciones duplicadas	Falta identificador local único o procesamiento idempotente en el central	Genera identificador en el dispositivo y haz que el servidor sea idempotente por ese identificador.
Sin conexión se realizan operaciones que no se pueden verificar y generan pérdidas	No hay lista escrita de qué se permite ni límites de riesgo	Decide con negocio qué opera sin conexión y con qué límites; lo no verificable, o se limita o se prohíbe.
Una actualización defectuosa obliga a visitar los dispositivos	Sin doble partición ni vuelta atrás automática, y desplegada a todos a la vez	Actualizaciones firmadas, instaladas en partición inactiva, con confirmación y reversión automática, y despliegue por grupos.

■ 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 aporta una SD-WAN que no dan los túneles configurados uno a uno?
2. ¿Qué aporta 5G de verdad y qué se le atribuye sin fundamento?
3. ¿Qué cinco cosas necesita una operación que debe funcionar sin conexión?
4. ¿Por qué no se resuelven los conflictos por la marca de tiempo del dispositivo?
5. ¿Qué diferencia hay entre sincronizar un contador por valor final y por variaciones?

Referencias

• Cisco (2025). SD-WAN design guide — política central y selección de camino por aplicación. <https://www.cisco.com/c/en/us/solutions/enterprise-networks/sd-wan/index.html>

• 3GPP (2025). 5G network slicing overview. <https://www.3gpp.org/technologies/5g-system-overview>

• AWS (2025). IoT Greengrass: local processing and offline operation. <https://docs.aws.amazon.com/greengrass/v2/developerguide/what-is-iot-greengrass.html>

• Shapiro, M. y otros (2011). Conflict-free replicated data types. <https://inria.hal.science/inria-00609399>

• Microsoft (2025). Azure IoT Edge deployment and safe rollout. <https://learn.microsoft.com/en-us/azure/iot-edge/about-iot-edge>

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

Descargar: Parte 16 en PDF · Manual integral

Anterior	Índice	Siguiente
← 202 · eBPF, flow logs, packet capture y diagnóstico	Parte 16 · Programa	204 · Proyecto: red multi-región y multi-cloud →

Evaluación — 203 SD-WAN, 5G, IoT y operación desconectada

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 edge-topology 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.

204 — Proyecto: red multi-región y multi-cloud

Parte: 16 — Redes cloud avanzadas, conectividad híbrida y edge Nivel: avanzado · Horas estimadas: 8 Laboratorio: capstone · Estado: EXECUTABLECORE

Propósito

Montar en un solo diseño lo de las once clases anteriores: direccionamiento, encaminamiento, nombres, entrada, borde, conectividad, salida, tráfico interno, diagnóstico y extremo, sobre varias regiones y varias nubes. La clase da el orden de decisión, el entregable y los criterios. Y cierra la parte 16: corrige las cinco predicciones de la clase 192 —tres acertadas, dos a medias—, actualiza el recuento de leyes, añade la ley 25 y escribe la hipótesis de la parte 17.

Resultados de aprendizaje

Al finalizar podrás:

1. Producir un diseño de red multirregión y multinube coherente.
2. Ordenar las decisiones de red por coste de cambio.
3. Comprobar el diseño con las pruebas negativas de toda la parte.
4. Corregir las cinco predicciones de la clase 192 con evidencia.
5. Escribir la hipótesis de la parte 17 en forma refutable.

Conceptos centrales

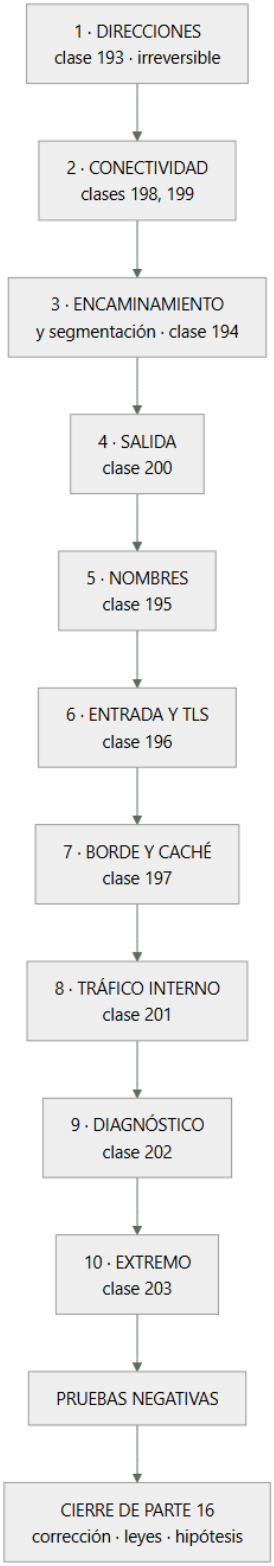
Concepto	Comprensión verificable
diseño de red completo	Conjunto coherente de decisiones desde el direccionamiento hasta el diagnóstico, sin huecos.
orden por coste de cambio	Direcciones primero, diagnóstico al final. Renumerar cuesta meses; añadir un panel, horas.
coherencia entre nubes	Que el plan de direcciones, los nombres y las políticas no se contradigan entre proveedores.
prueba negativa de red	Comprobación que provoca el fallo de red a propósito y verifica la respuesta declarada.
ley 25	Lo provisional sobrevive a su motivo; sin fecha de caducidad, para siempre.
hipótesis de parte	Afirmación refutable escrita antes de estudiar, que la parte siguiente corrige con evidencia.

Modelo mental

La red cloud es un sistema distribuido de rutas, identidades y políticas; cada salto añade latencia, costo y una frontera de fallo.

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

■ Flujo de razonamiento



Desarrollo

1. El encargo y su orden

El encargo. CloudShop opera en tres nubes, dos regiones por nube, un centro de datos, once oficinas y trescientas cuarenta tiendas. El proyecto consiste en producir el diseño de red completo.

El orden de decisión, por coste de cambio:

```
1  DIRECCIONAMIENTO                              clase 193
   espacio reservado, jerarquía, registro central
   → lo más caro de cambiar de todo el programa

2  CONECTIVIDAD FÍSICA Y LÓGICA            clases 198, 199
   enlaces y túneles con su redundancia sin causa común
   concentradores por región, unidos entre sí

3  ENCAMINAMIENTO Y SEGMENTACIÓN                 clase 194
   tablas por función; cerrar por ausencia de ruta
   anuncios resumidos y filtros explícitos

4  CONTROL DE SALIDA                             clase 200
   puntos privados, denegación por defecto, perímetro de
   datos

5  NOMBRES                                       clase 195
   zonas, vista partida, TTL atados al plazo de recuperación

6  ENTRADA Y CERTIFICADOS                        clase 196
   capas 4 y 7, reparto, salud, renovación automática

7  BORDE Y CACHÉ                                 clase 197
   clave de caché, rancio, escudo de origen

8  TRÁFICO INTERNO                               clase 201
   identidad de carga y autorización, con lo mínimo

9  DIAGNÓSTICO                                   clase 202
   flujos sin muestreo, observación del núcleo, captura
   preautorizada

10 EXTREMO                                       clase 203
   qué opera sin conexión, con qué límites y cómo reconcilia
```

Y las dos reglas del orden:

```
los pasos 1 y 2 condicionan todo lo demás
los pasos 9 y 10 se pueden cambiar en semanas
→ y por eso la discusión debe gastarse en los primeros
```

Y el error de método que hunde este proyecto:

```
empezar por el paso 6 o el 7, que son los visibles
→ y descubrir en el paso 1 que hay solapamiento y que nada
  de lo anterior se puede conectar                clase 193
```

2. El entregable y las pruebas

El documento, en unas quince páginas:

```
1  PLAN DE DIRECCIONAMIENTO
   espacio, jerarquía, reservas y registro
   consumidores contados, incluidos pods y puntos privados

2  TOPOLOGÍA
   enlaces y túneles, con ruta física documentada
   concentradores y su unión entre regiones y nubes
   qué anuncia y qué acepta cada extremo

3  SEGMENTACIÓN
   tablas de rutas por función
   qué alcanza cada segmento, y qué NO por ausencia de ruta

4  SALIDA
   puntos privados, destinos permitidos con dueño y
   caducidad, perímetro de datos

5  NOMBRES
   zonas, vista partida, TTL por volatilidad, resolución
   entre nubes y corporativa

6  ENTRADA
   capas, algoritmos, comprobaciones, plazos ordenados
   inventario de certificados y su renovación
```

```
7  BORDE
   claves de caché por ruta, políticas de rancio, escudo

8  TRÁFICO INTERNO
   identidad, autorización, y qué capacidades NO se adoptan

9  DIAGNÓSTICO
   qué se registra, sin muestreo dónde, y procedimiento de
   captura

10 EXTREMO
   operaciones sin conexión, límites y reconciliación

11 CAMINOS ESPERADOS de los flujos principales  clase 194
12 PRUEBAS NEGATIVAS con su resultado
13 LO QUE NO SE HACE, y por qué
```

Las pruebas negativas de la parte, que son el criterio de terminado:

- superar el límite de prefijos de una sesión y ver la caída
- trazar ida y vuelta de los flujos principales
- crear una ruta a un rango y comprobar que no se propaga donde no debe
- alcanzar producción desde no-producción
- sacar datos a un almacén de otra organización
- sacar datos por consultas de nombres
- cambiar un registro y medir cuánto tarda cada servicio
- enviar un paquete grande sin fragmentar por cada túnel
- desconectar el enlace principal y cronometrar la vuelta
- dejar caducar un certificado de prueba y ver la alerta
- degradar una réplica y ver si el reparto la evita
- invalidar caché en masa y medir la carga del origen
- inyectar fallo en una dependencia declarada blanda
- desconectar el plano de control de la SD-WAN
- cortar la luz a un dispositivo a mitad de operación

Y los criterios de evaluación, publicados antes:

```
1  el plan de direcciones cuenta los consumidores ocultos  3
2  hay registro central obligatorio y automatizado         2
3  la redundancia no tiene causa común, documentada        3
4  la segmentación cierra por ausencia de ruta             3
5  la salida está cerrada y el perímetro de datos existe   3
6  los TTL están atados al plazo de recuperación           2
7  los certificados tienen alerta por antigüedad           2
8  la clave de caché está justificada por ruta             2
9  solo se adoptan las capacidades de malla que faltan     2
10 los flujos sin muestreo en lo crítico                   1
11 está escrito qué opera sin conexión y con qué límites   3
12 están escritos los caminos esperados                    2
13 las pruebas negativas se ejecutaron y hay fallos
   publicados                                              3
```

Y el 13 pesa como los que más, por el mismo motivo de siempre:

```
en este programa, la proporción de pruebas negativas que
fallan la primera vez ha estado entre el 27 % y el 45 %
→ un proyecto con cero fallos no las ejecutó       ley 22
```

3. Cierre de la parte 16: corrección de las cinco predicciones

Las cinco predicciones de la clase 192, corregidas con la evidencia de las clases 193 a 203.

1. «la red será la capa donde más decisiones irreversibles se toman con menos deliberación: los rangos y la conectividad se eligen en una tarde y condicionan una década»

   CORRECTA, y con las cifras más contundentes de la parte. 87 redes, 37 sin registro alguno, 10.0.0.0/16 usado en 6 sitios a la vez, y 172.17 solapando con un motor de contenedores desde 2022. Reservar el espacio y registrar los bloques al principio costaba 2 días; renumerar una sola red costó 25 personas-semana, y el plazo no lo marcó ninguna decisión técnica sino la lista de permitidos de un socio. Ley 14 en su forma más cara.

2. «la mayoría de los problemas de red no serán de encaminamiento ni de rendimiento sino de nombres y certificados»

   A MEDIAS, y la cuenta lo aclara. De los diecisiete incidentes de la parte: nombres 5, certificados 2, encaminamiento 5, y el resto repartido entre MTU, redundancia falsa, reparto, exfiltración y reconciliación. Nombres y certificados suman 7 de 17 (41 %): la familia más grande, pero no la mayoría, y empatados con encaminamiento en incidentes puros. Donde sí acertamos de pleno es en la GRAVEDAD: las dos únicas caídas TOTALES del año fueron de certificado, y ninguna por caducidad sorpresa sino por automatización rota en silencio.

3. «el coste de salida y el tráfico entre zonas volverán a ser la línea más grande y la peor atribuida, y seguirá sin tener dueño»

   PRIMERA MITAD CORRECTA: el 82 % del tráfico saliente iba a servicios del propio proveedor pagándose como salida a internet, y no lo miraba nadie. SEGUNDA MITAD FALLADA, y por un motivo que no habíamos previsto: cerrar la salida por seguridad OBLIGÓ a declarar cada destino con dueño y caducidad. Al final del año había 57 destinos con dueño donde antes había una salida abierta sin ninguno. Predijimos que seguiría sin dueño y le puso dueño un proyecto de seguridad, no uno de coste.

4. «la malla aparecerá otra vez como respuesta a problemas ya resueltos en otra capa, y el análisis honesto la dejará en una o dos capacidades»

   CORRECTA Y EXACTA: de las cinco capacidades, tres estaban resueltas, se adoptó por dos —identidad y autorización— y en el modo más barato. Pero incompleta en algo interesante: la capacidad que más problemas reales destapó no fue ninguna de las dos que justificaron la adopción, sino la inyección de fallos, que reveló que dos de cinco dependencias declaradas blandas eran duras.

5. «el diagnóstico de red será donde más se note la ley 24: los diagramas omitirán lo mismo, y ahí estarán los incidentes»

   CORRECTA Y SUBESTIMADA. Un camino de desarrollo a los datos de producción abierto tres años; 1.334 destinos externos desconocidos, entre ellos una exfiltración de catorce meses al almacén personal de un antiguo empleado; 96 llamadas entre servicios donde el diagrama declaraba 61; un registro de nombres huérfano de diecinueve meses. Las cuatro omisiones que la ley 24 nombra aparecieron las cuatro, y ninguna la detectó una alerta: las detectaron inventarios.

Marcador: tres correctas, dos a medias. Y por primera vez en el programa el fallo no fue de reparto sino de mecanismo: en la predicción 3 dimos por hecho que un problema de coste sin dueño seguiría sin dueño, y lo resolvió un proyecto de otra disciplina. Los problemas no siempre los arregla quien los sufre.

4. Recuento de leyes, ley 25 e hipótesis de la parte 17

El recuento de leyes, cerrada la parte 16.

ley 13	lo que no se mira deja de funcionar en silencio	39
ley 15	la señal existe y nadie la mira	29
ley 14	el coste se decide al crear, no al pagar	25
ley 22	un procedimiento nunca ejecutado no funciona	24
ley 16	un control que estorba se rodea	23
ley 20	lo que no tiene dueño se filtra y se desperdicia	21
ley 21	el acoplamiento vive en quién escribe	18
ley 19	la compensación hace invisible el fallo	10
ley 17	se optimiza la medida, no el objetivo	10
ley 23	la capacidad la limita lo que ya se mantiene	10

```
ley 24  lo que no está en el diagrama no se analiza            9
ley 18  lo asíncrono traslada la garantía, no la elimina       8
```

Y la parte 16 obliga a escribir una ley nueva, que este programa lleva rozando desde la parte 08:

```
LEY 25
  lo provisional sobrevive a su motivo;
  sin fecha de caducidad, para siempre

apariciones en esta parte                                      5
  clase 194   una preferencia cambiada «para probar el
              respaldo» que llevaba 2 semanas rompiendo la
              simetría; una ruta a agujero negro de una
              migración de agosto
  clase 195   un registro de nombres huérfano de 19 meses
  clase 199   31 emparejamientos creados «para una prueba»;
              rutas estáticas de una migración terminada en
              2022 que abrían desarrollo a producción
  clase 200   un agente de monitorización de un contrato
              cancelado, aún enviando datos; un script de
              exportación de un empleado que ya no estaba
  clase 203   nada excluido de la operación sin conexión,
              porque nunca se decidió qué excluir

y lo que la distingue de la ley 20
  la 20 dice que lo sin dueño se degrada
  la 25 dice que lo temporal NO se retira, tenga dueño o no
  → el remedio no es asignar dueño: es poner fecha
```

La hipótesis de la parte 17 (clases 205 a 216, AWS en profundidad y proyecto productivo), escrita antes de estudiarla para que la clase 216 la corrija:

1. bajar a un proveedor concreto va a revelar que buena parte de lo que este programa ha tratado como decisiones de arquitectura son, en ese proveedor, valores por defecto; y los valores por defecto van a resultar mal elegidos para producción en más de la mitad de los casos

2. el diseño de la base de datos por patrones de acceso será la decisión más cara de cambiar de toda la parte, y la que más veces se tome sin haber escrito los patrones
   ley 14

3. la eliminación de secretos mediante federación resultará sencilla técnicamente y lenta organizativamente: el obstáculo no será la configuración sino quién tiene permiso para cambiarla
   ley 16

4. en el proyecto productivo, más de la mitad de los problemas que aparezcan no serán de AWS sino de lo mismo de siempre: algo provisional que no se retiró, una señal que nadie miraba y un procedimiento nunca ejecutado
   leyes 25, 15, 22

5. el coste real del sistema terminado será entre dos y cuatro veces la estimación inicial, y la diferencia estará casi toda en partidas que no son cómputo: transferencia, almacenamiento de registros y servicios gestionados facturados por petición

Y el cierre de la parte 16: de once clases, lo que más problemas destapó no fue ninguna herramienta de diagnóstico, sino tres inventarios —el de bloques de direcciones, el de destinos de salida y el de llamadas entre servicios—. Ninguno de los tres requería tecnología nueva; los tres encontraron cosas que llevaban años funcionando sin que nadie las hubiera dibujado. La parte 17 baja a un proveedor concreto y construye un sistema productivo entero, empezando por el alojamiento y la entrega. Es la clase 205.

Ejemplo trabajado

El diseño de red de CloudShop, resuelto con el método de la parte. Lo que sigue es el resumen del entregable con las decisiones que costaron discusión, y el resultado de las quince pruebas negativas —de las que fallaron seis.

Paso 1 · Direccionamiento.

```
espacio reservado                          10.0.0.0/8
jerarquía                region /12 · entorno /16 ·
                         dominio /18 · zona /20 · función /24
reservados a propósito
  10.80.0.0/12   adquisiciones
  10.96.0.0/11   libre
pods en espacio no enrutado                100.64.0.0/10
  → esta decisión sola evitó pasar de /22 a /18 por zona

registro central obligatorio, automatizado en la plantilla
tiempo para obtener un bloque                    40 s
función de aptitud   ninguna red sin bloque registrado
```

Paso 2 · Conectividad.

```
centro de datos ↔ nube principal
  2 enlaces dedicados de 1 Gbps
  proveedores de última milla DISTINTOS
  edificios de entrada DISTINTOS
  ruta física documentada por escrito, tras dos semanas de
  insistir                                        clase 198
  túnel de respaldo, ejercitado mensualmente

concentradores
  uno por región y por nube: 6 en total
  emparejados entre sí
  63 redes conectadas
  2 pares de gran volumen FUERA del concentrador
    → 11.840 €/mes evitados                       clase 199

tiendas
  SD-WAN, 2 enlaces por tienda (fibra + 5G)
  política central, autonomía si el plano no responde
```

Paso 3 · Segmentación.

```
tablas de rutas
  producción · no-producción · compartidos · socios ·
  inspección

lo que NO se alcanza, por ausencia de ruta
  no-producción → producción
  socios → cualquier cosa que no sea la red de intercambio
  datos → internet

anuncios
  la nube anuncia 2 prefijos resumidos, no 340
  la nube acepta 31 prefijos filtrados, no 1.043
  alerta al 75 % del límite
```

Paso 4 · Salida.

```
27 puntos privados centralizados en red compartida
cortafuegos de salida con denegación por defecto
57 destinos permitidos, cada uno con dueño y caducidad
perímetro de datos en las dos direcciones
resolución forzada por el resolutor propio

coste                    9.450 €/mes → 4.250 €/mes
```

Pasos 5 a 10, resumidos con lo que costó discusión:

```
NOMBRES
  subdominio interno int.cloudshop.com → vista partida solo
  en 3 nombres
  TTL por volatilidad; 30 s en lo que conmuta
  DNSSEC evaluado y NO adoptado, con registro    clase 195

ENTRADA
  capa 4 con IP fija delante (listas de terceros)
  capa 7 detrás: TLS con recifrado, rutas, plazos ordenados
  reparto por menor número de conexiones + expulsión de
  atípicos
  31 certificados, todos automáticos, alerta por antigüedad

BORDE
```

```
  claves de caché con lista blanca por ruta
  rancio mientras refresca y si hay error
  invalidación por etiqueta; comodín prohibido
  escudo de origen
  aciertos globales                        71 % → 97,2 %

TRÁFICO INTERNO
  malla en modo por nodo, SOLO identidad y autorización
  reintentos y plazos siguen en la biblioteca
  alcance desde un servicio comprometido      11 → 2

DIAGNÓSTICO
  flujos sin muestreo en producción y datos
  observación del núcleo en todos los nodos
  captura preautorizada, con caducidad de 7 días
  panel de RECHAZOS revisado semanalmente

EXTREMO
  lista escrita de operaciones sin conexión, con límites
  cola persistente idempotente
  reglas de reconciliación por tipo de dato
  actualizaciones con doble partición y vuelta atrás
```

Las quince pruebas negativas: seis fallaron.

```
✓  superar el límite de prefijos → sesión cae, alerta salta
✓  trazar ida y vuelta de los 6 flujos principales
✗  ruta que no debe propagarse
   → se propagó a la tabla de compartidos por una asociación
     mal configurada al añadir la sexta región
✓  alcanzar producción desde no-producción → sin ruta
✗  sacar datos al almacén de otra organización
   → funcionó: faltaba política en 4 de los 27 puntos
     privados                                      clase 200
✗  sacar datos por consultas de nombres
   → funcionó desde las tiendas: el resolutor forzado se
     aplicó en la nube y no en la SD-WAN
✓  cambiar un registro y medir seguimiento → 50 s máximo
✗  paquete grande sin fragmentar por cada túnel
   → 3 de 340 tiendas fallaron; equipos con configuración
     antigua
✓  desconectar el enlace principal → 52 s
✓  certificado de prueba caducado → alerta a los 4 min
✓  degradar una réplica → retirada en 35 s
✓  invalidar en masa → prohibido por política; se comprobó
   que la API lo rechaza
✗  inyectar fallo en dependencia blanda
   → 2 de 5 eran duras                            clase 201
✓  desconectar el plano de control de la SD-WAN 4 h
✗  cortar la luz a mitad de operación
   → 1 de 20 se perdió: un modelo de dispositivo no forzaba
     la sincronización a disco
```

Y el análisis de las seis:

```
dos por aplicación incompleta de un control (puntos
  privados, resolutor forzado)
dos por equipos o dispositivos con configuración antigua
una por propagación mal configurada al crecer
una por una premisa no comprobada (dependencias blandas)

→ ninguna por un error de diseño
→ las seis por la distancia entre lo diseñado y lo
  desplegado                                     ley 22
```

Lo que se decidió no hacer:

```
no adoptar DNSSEC: modo de fallo apaga el dominio y el
  riesgo está cubierto por TLS en lo que importa
no adoptar la malla completa: tres capacidades ya resueltas
no inspeccionar el tráfico dentro de un mismo segmento:
  triplica coste sin ganar nada
no montar segunda región activa para el flujo de compra:
  6.400 €/mes frente a 390 € de pérdida esperada  clase 185
no unificar el proveedor de DNS entre nubes: el riesgo del
```

```
cambio supera la ventaja; revisar en 12 meses
```

La lección que este proyecto deja: de las quince pruebas negativas, seis fallaron y ninguna por un error del diseño. Todas por la distancia entre lo que estaba escrito y lo que estaba desplegado: cuatro puntos privados sin política, un resolutor forzado que solo se aplicó en una de las dos topologías, tres tiendas con configuración vieja y un modelo de dispositivo que no sincronizaba a disco. El diseño se puede revisar leyendo; lo desplegado, solo rompiéndolo.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-16-advanced-cloud-networking-edge/204-proyecto-red-multi-region-y-multi-cloud/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 multicloud-network 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 multicloud-network 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
Nada de lo diseñado se puede conectar entre sí	Se empezó por las capas visibles y el direccionamiento tenía solapamientos	Decide el plan de direcciones y el registro central antes que ninguna otra cosa de red.
Un control se aplica en una topología y no en la otra	El diseño se validó leyendo y no ejecutando	Ejecuta cada prueba negativa desde cada tipo de red: nube, corporativa y sedes.
La redundancia contratada no protege del fallo más probable	Causa común física o lógica no comprobada	Exige por escrito la ruta física de cada circuito y revisa también las causas comunes lógicas, como el límite de prefijos.
Al crecer, aparecen caminos que no deberían existir	Propagación mal configurada al añadir una región o una red	Declara qué prefijos deben verse desde cada tabla y alerta ante cualquier ruta inesperada.
El proyecto declara todas las pruebas superadas	No se ejecutaron: en este programa fallan entre el 27 % y el 45 % la primera vez	Ejecuta y publica los fallos; un resultado sin fallos indica que la prueba no se hizo.
Se acumulan rutas, reglas y excepciones que nadie recuerda por qué existen	Se crearon como provisionales y sin fecha de caducidad	Toda excepción, ruta manual o emparejamiento temporal nace con dueño y fecha; sin fecha, no se crea.

Seguridad, ética y costo

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

Preguntas de comprobación

1. ¿Por qué el direccionamiento se decide antes que cualquier otra cosa de red?
2. ¿Cuál de las cinco predicciones de la clase 192 falló y por qué motivo nuevo?
3. ¿Qué dice la ley 25 y en qué se distingue de la ley 20?
4. ¿Qué proporción de pruebas negativas ha fallado la primera vez en este programa?
5. ¿Qué encontraron los tres inventarios de esta parte que ninguna alerta detectó?

Referencias

- AWS (2025). Building a scalable and secure multi-VPC network infrastructure. <https://docs.aws.amazon.com/whitepapers/latest/building-scalable-secure-multi-vpc-network-infrastructure/welcome.html>
- Microsoft (2025). Cloud Adoption Framework: network topology and connectivity. <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/landing-zone/design-area/network-topology-and-connectivity>
- Google Cloud (2025). Network design decisions in the Architecture Framework. <https://cloud.google.com/architecture/framework>
- Beyer, B. y otros (2018). The Site Reliability Workbook — verificación con pruebas reales. <https://sre.google/workbook/table-of-contents/>
- NIST SP 800-207 (2020). Zero Trust Architecture. <https://csrc.nist.gov/pubs/sp/800/207/final>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 16 en PDF · Manual integral

Anterior	Índice	Siguiente
← 203 · SD-WAN, 5G, IoT y operación desconectada	Parte 16 · Programa	205 · Hosting progresivo con Amplify, S3 y CloudFront →

Evaluación — 204 Proyecto: red multi-región y 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 multicloud-network 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.