Multi-Cloud Engineering Program

Parte 18 — Azure: arquitectura empresarial y operación en producción

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	18 de 23
Clases	217–228
Evaluaciones	12
Horas estimadas	52

Índice

Parte 18 — Azure: arquitectura empresarial y operación en producción

← Parte 17 · Índice completo · Parte 19 →

Descargar: Esta parte en PDF · Recorrido de Azure en PDF · Manual integral

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

Operar CloudShop sobre landing zones, PaaS, datos, identidad y observabilidad de Azure.

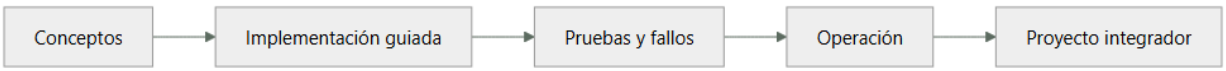
Resultados de la parte

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 arquitectura Azure nace del tenant y las políticas heredables, y llega al workload mediante identidades administradas y redes privadas.

Secuencia



Clases

ID	Clase	Laboratorio	Horas
217	Enterprise-scale landing zones y management groups	governance	4
218	Entra ID, workload identity, PIM y Conditional Access	iam	4
219	Hub-spoke, Virtual WAN, Private Link y DNS privado	network	4
220	Bicep, deployment stacks y Azure Verified Modules	iac	4
221	App Service, Functions y Container Apps en producción	serverless	4
222	AKS, workload identity, ingress y GitOps	kubernetes	4
223	Azure SQL, Cosmos DB y consistencia distribuida	data	4
224	Service Bus, Event Grid y Event Hubs	messaging	4
225	Azure Monitor, Application Insights y OpenTelemetry	observability	4
226	Defender for Cloud, Policy y Sentinel	security	4
227	Cost Management, Advisor, resiliencia y Chaos Studio	finops	4
228	Proyecto: CloudShop productivo en Azure	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

- Azure Architecture Center — Microsoft.
- Exam Ref AZ-305 — Microsoft Press.
- Designing Distributed Systems — Brendan Burns.

217 — Enterprise-scale landing zones y management groups

Parte: 18 — Azure: arquitectura empresarial y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: governance · Estado: EXECUTABLECORE

Propósito

Diseñar la jerarquía de Azure —grupos de administración, suscripciones y grupos de recursos— que es a esta nube lo que el plan de direcciones a la red: se decide en una tarde y condiciona una década. La clase da el criterio de reparto, explica dónde se aplican los controles para que no se puedan quitar desde abajo, y desarrolla la parte que más cuesta corregir después: mover una suscripción de sitio no mueve lo que ya se creó con la política antigua.

Resultados de aprendizaje

Al finalizar podrás:

1. Repartir cargas entre grupos de administración y suscripciones con criterio.
2. Aplicar controles en el nivel correcto, sin que se puedan quitar desde abajo.
3. Distinguir política, iniciativa y bloqueo, y saber qué resuelve cada uno.
4. Desplegar políticas en modo auditoría antes de denegar.
5. Corregir una jerarquía mal repartida sin romper lo existente.

Conceptos centrales

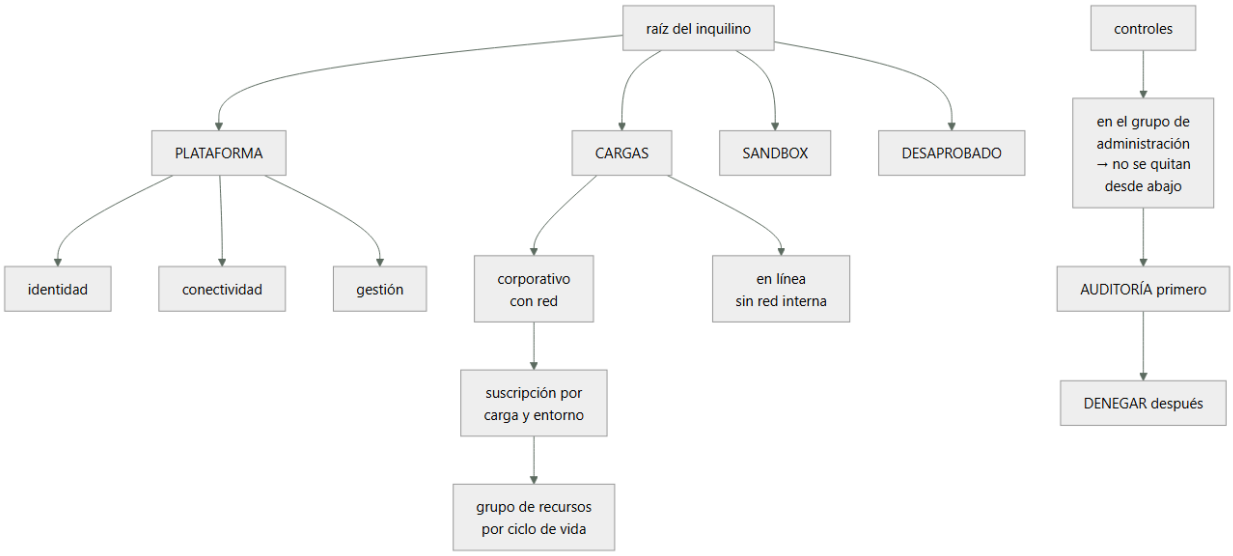
Concepto	Comprensión verificable
grupo de administración	Contenedor de suscripciones donde se aplican políticas y permisos heredados. Es el nivel que fija el gobierno.
suscripción	Unidad de facturación, de cuota y de aislamiento. Equivale funcionalmente a una cuenta.
grupo de recursos	Agrupación dentro de una suscripción, con ciclo de vida común. Se borra entero.
política	Regla que audita, deniega, modifica o despliega. Se hereda hacia abajo y no se puede quitar desde abajo.
iniciativa	Conjunto de políticas asignado como una unidad, con parámetros. Es la forma práctica de gobernar.
zona de aterrizaje	Suscripción preparada con red, identidad, controles y observabilidad, lista para recibir cargas.

Modelo mental

La arquitectura Azure nace del tenant y las políticas heredables, y llega al workload mediante identidades administradas y redes privadas.

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 jerarquía, y por qué se decide pronto

Azure tiene tres niveles y cada uno resuelve algo distinto. Confundirlos es el error inicial.

```
GRUPO DE ADMINISTRACIÓN
  contenedor de suscripciones
  aquí van las POLÍTICAS y las asignaciones de permisos que
  se heredan
  → es el nivel del gobierno

SUSCRIPCIÓN
  unidad de facturación, de CUOTA y de aislamiento
  → los límites de recursos son por suscripción, y ese es
    el motivo práctico más frecuente para separar

GRUPO DE RECURSOS
  agrupación con ciclo de vida común
  → se borra entero; los recursos de un grupo deberían
    nacer y morir juntos
  → no es un mecanismo de aislamiento de seguridad fuerte
```

Y el reparto que funciona, que conviene copiar salvo buena razón:

```
raíz
■■■ PLATAFORMA
■    ■■■ identidad          directorio, servidores heredados
■    ■■■ conectividad       red central, cortafuegos, DNS
■    ■■■ gestión            registros, copias, herramientas
■■■ CARGAS
■    ■■■ corporativo        necesita red interna
■    ■■■ en línea           público, sin red interna
■■■ SANDBOX                 experimentación, con caducidad
■■■ DESAPROBADO             lo que se retira, con controles
                            más duros y sin crecimiento
```

Y las tres decisiones que hay que tomar y que cuestan corregir:

```
1  ¿UNA SUSCRIPCIÓN POR ENTORNO O POR CARGA?
   por entorno y por carga: producción de pedidos, una;
   preproducción de pedidos, otra
   → aísla cuota, facturación y permisos a la vez
   → y es lo que permite dar acceso de administrador a un
     equipo sobre SU entorno sin exponer los demás

2  ¿DÓNDE VAN LOS CONTROLES?
   en el grupo de administración, nunca en la suscripción
   → una política asignada en la suscripción la puede
     quitar quien administre esa suscripción
   → una asignada arriba, no                    clase 169
```

```
3  ¿QUÉ SE HEREDA Y QUÉ SE PARAMETRIZA?
   la misma iniciativa arriba, con parámetros distintos por
   rama
   → «regiones permitidas» puede ser más estricto en
     corporativo que en sandbox
```

Y la advertencia que da nombre a la clase:

```
MOVER UNA SUSCRIPCIÓN DE SITIO NO ARREGLA LO YA CREADO
  las políticas de denegación solo actúan sobre
  CREACIONES y MODIFICACIONES
  → los recursos que ya existen siguen incumpliendo
  → aparecen como no conformes, y hay que corregirlos uno a
    uno o con tareas de remediación

→ y por eso la jerarquía se decide antes de crear nada
                                                    ley 14
```

2. Políticas: auditar antes de denegar

Las políticas son el mecanismo de gobierno de Azure y tienen cuatro efectos, con usos distintos.

```
AUDITAR      registra el incumplimiento, no impide nada
             → siempre el primer paso
DENEGAR      impide crear o modificar
             → el control real
MODIFICAR    corrige el recurso al crearlo (añade etiquetas,
             activa cifrado)
             → mejor que denegar cuando se puede arreglar
               solo                              clase 214
DESPLEGAR SI NO EXISTE
             crea lo que falta (diagnóstico, agente)
             → potente y peligroso: despliega cosas que
               nadie pidió y que facturan
```

Y el orden de implantación, que es el mismo de la clase 200:

```
1  ASIGNAR EN MODO AUDITORÍA, en el grupo de administración
2  MEDIR 2-4 semanas: cuántos recursos incumplen y cuáles
3  CLASIFICAR: incumplimientos reales frente a excepciones
   legítimas
4  CORREGIR o EXCEPTUAR, con dueño y fecha
5  CAMBIAR A DENEGAR
6  VIGILAR el cumplimiento como una señal más   clase 211

→ asignar en denegar el primer día bloquea despliegues
  legítimos y el gobierno pierde credibilidad     ley 16
```

Las iniciativas, que es como se gobierna de verdad:

```
una política suelta es difícil de mantener
una INICIATIVA agrupa decenas con parámetros
  → se asigna una vez por rama, con sus parámetros
  → y el cumplimiento se mide por iniciativa, no política a
    política

las iniciativas mínimas de una organización
  etiquetas obligatorias y sus valores          clase 214
  regiones permitidas
  tipos de recurso permitidos o prohibidos
  cifrado obligatorio y versión mínima de TLS
  sin acceso público en almacenamiento ni en bases
  diagnóstico enviado al destino central
  y las de cumplimiento normativo, si aplican
```

Y una advertencia sobre el modo de despliegue automático:

```
una política que despliega el agente de diagnóstico en
cada recurso nuevo
  → resuelve el problema de la observabilidad ausente
  → y crea coste que nadie pidió                clase 214
→ hay que estimar el coste de lo que la política despliega
  ANTES de asignarla
```

Los bloqueos, que son otra cosa y se confunden con las políticas:

```
BLOQUEO DE BORRADO      impide eliminar el recurso
```

```
BLOQUEO DE SOLO LECTURA impide modificarlo

  se aplican al recurso o al grupo, y los hereda lo de dentro
  → útiles para lo que nunca debe borrarse: bases de datos
    de producción, red central, copias
  → y son un incordio si se ponen de más: bloquean el
    despliegue declarativo            ← y entonces se quitan
                                        y nadie los repone
                                                  ley 25
```

3. Suscripciones: cuotas, facturación y aislamiento

La suscripción es el nivel donde se topan los límites, y ese es el motivo práctico que más veces obliga a separar.

```
LO QUE ES POR SUSCRIPCIÓN
  cuotas de cómputo por familia y por región
  límites de recursos por tipo
  la factura y su desglose
  y el ámbito natural de muchos permisos

CONSECUENCIAS PRÁCTICAS
  una carga que consume mucha cuota puede impedir crecer a
  otra de la misma suscripción
  → separar por carga aísla ese riesgo

  las cuotas NO son automáticas: hay que pedirlas con
  antelación
  → y en un incidente que requiera escalar mucho, la cuota
    es el límite real, no la capacidad de Azure
  → por eso se pide margen antes, y se vigila el consumo
    frente al límite                          clase 262
```

Las zonas de aterrizaje, que es la forma de que una suscripción nueva nazca lista:

```
QUÉ TRAE UNA SUSCRIPCIÓN ENTREGADA
  colocada en el grupo de administración correcto
  red conectada al centro, con sus rangos del plan
                                            clase 219
  identidad y permisos base asignados
  diagnóstico enrutado al destino central
  presupuesto y etiquetas obligatorias      clase 214
  y las políticas heredadas ya aplicando

Y EL PLAZO IMPORTA
  si entregar una suscripción tarda tres semanas, los
  equipos usarán la suya, o crearán recursos donde puedan
                                                  ley 16
  → automatizar la entrega es lo que hace que el gobierno
    se cumpla                                clase 171
```

Y la decisión sobre quién administra qué:

```
EL EQUIPO DE PLATAFORMA administra
  grupos de administración, políticas, red central,
  identidad, destinos de diagnóstico

EL EQUIPO DE LA CARGA administra
  su suscripción: crea, despliega y opera dentro de las
  reglas heredadas

Y LO QUE NADIE DEBE PODER HACER
  quitar una política heredada
  cambiar el enrutamiento del diagnóstico
  crear emparejamientos de red por su cuenta
  → estas son las que van arriba, denegadas
```

4. Corregir una jerarquía mal repartida

Casi ninguna organización empieza con la jerarquía correcta. Corregirla se puede, y hay un orden.

```
LO QUE SE PUEDE MOVER
  una suscripción puede cambiar de grupo de administración
  un recurso puede cambiar de grupo de recursos (a veces)
  y en algunos casos, de suscripción
```

```
LO QUE NO SE MUEVE FÁCIL
  recursos con dependencias de red
  recursos con identidad administrada asignada
  claves y secretos con referencias
  y todo lo que tenga la suscripción escrita en su
  identificador en otro sitio
```

Y el procedimiento:

```
1  DEFINIR la jerarquía objetivo y las iniciativas
2  ASIGNARLAS EN AUDITORÍA en el nivel correcto
3  MEDIR el incumplimiento actual
     → esta es la fotografía que dice cuánto trabajo hay
4  MOVER las suscripciones al grupo que les corresponde
     → recordando que lo ya creado sigue incumpliendo
5  REMEDIAR lo existente, por lotes y por prioridad
     → primero lo que afecta a seguridad y a datos
6  PASAR A DENEGAR cuando el incumplimiento sea residual
7  DEJAR el cumplimiento como señal vigilada
```

Y lo que hay que medir mientras tanto:

```
recursos conformes frente a totales, por iniciativa
y su TENDENCIA
  → si no baja, la remediación no está ocurriendo
excepciones vivas y su antigüedad              clase 190
  → si crecen siempre, la política está mal calibrada
```

Y una advertencia sobre las exenciones:

```
una exención sin fecha es una política que no existe
                                                    ley 25
→ toda exención con dueño, motivo y caducidad
→ y un panel con las que vencen este mes
```

Y la lista de comprobación de la clase:

- hay grupos separados para plataforma, cargas, sandbox y desaprobado
- las suscripciones se separan por carga Y por entorno
- todas las políticas se asignan en grupos de administración, no en suscripciones
- las políticas se asignaron primero en auditoría
- se midió el incumplimiento antes de denegar
- las iniciativas están parametrizadas por rama
- está estimado el coste de lo que despliegan las políticas
- los bloqueos están solo donde hacen falta y no estorban al despliegue declarativo
- las cuotas están pedidas con margen y se vigilan
- entregar una suscripción nueva está automatizado y tarda minutos
- ningún equipo de carga puede quitar una política heredada
- toda exención tiene dueño, motivo y caducidad
- el cumplimiento por iniciativa se vigila como una señal

Y el cierre que enlaza con la clase siguiente: con la jerarquía puesta, el control que de verdad decide el alcance en Azure no es la política sino quién puede hacer qué y desde dónde. Identidad, identidades de carga, elevación temporal y acceso condicional es la materia de la clase 218.

Ejemplo trabajado

CloudShop lleva cuatro años en Azure con 61 suscripciones creadas sin criterio. Lo que sigue es la fotografía del incumplimiento, la jerarquía nueva, y el hallazgo de que mover las suscripciones no arregló casi nada.

El punto de partida:

```
suscripciones                                  61
  bajo la raíz, sin grupo de administración    47
  en un grupo llamado «Producción»              9
  en un grupo llamado «Test»                    5

políticas asignadas                            12
  en grupos de administración                   3
  EN SUSCRIPCIONES                              9  ←
  de las 9, quitadas por el equipo de la carga  4
```

```
y la consecuencia
  4 equipos habían quitado las políticas de su propia
  suscripción porque «bloqueaban el despliegue»
  → y nadie se enteró: no había alerta de cambio de
    asignación                                          ley 15
```

Y el detalle de por qué las quitaron:

```
la política denegaba crear cuentas de almacenamiento sin
cifrado con clave propia
la plantilla del equipo no lo declaraba
y el mensaje de error decía solo «denegado por política»
→ nadie sabía qué faltaba
→ y quitar la política era más rápido que averiguarlo
                                                      ley 16
```

La jerarquía nueva:

```
raíz
■■■ plataforma
■    ■■■ identidad          1 suscripción
■    ■■■ conectividad       2 (una por región)
■    ■■■ gestión            1
■■■ cargas
■    ■■■ corporativo        34 suscripciones
■    ■     pedidos-prod, pedidos-pre, pedidos-dev,
■    ■     catalogo-prod, … (una por carga y entorno)
■    ■■■ en línea            9
■■■ sandbox                12  ← con caducidad de 90 días
■■■ desaprobado             2

y el criterio escrito
  una suscripción por CARGA y por ENTORNO
  motivo   aísla cuota, factura y permisos a la vez
  coste de cambio si nos equivocamos   alto
```

Las iniciativas, con sus parámetros por rama:

```
iniciativa                    corporativo  en línea  sandbox
etiquetas obligatorias           denegar    denegar   auditar
regiones permitidas              2 (UE)     2 (UE)    2 (UE)
sin acceso público                denegar    denegar   denegar
cifrado en reposo                 denegar    denegar   auditar
TLS mínimo 1.2                    denegar    denegar   denegar
diagnóstico al destino central   desplegar  desplegar auditar
tipos de recurso permitidos      restringido  amplio   amplio
sin IP pública en máquinas        denegar    auditar   auditar
```

Y el coste que se estimó antes de asignar:

```
la política que despliega el diagnóstico crearía
  un ajuste de diagnóstico por recurso
  ingesta estimada a partir del inventario  ~1.900 €/mes
→ se ajustó qué categorías se envían y con qué caducidad
→ ingesta real                                640 €/mes
→ sin esta estimación previa, habría sido una sorpresa en
  la factura del mes siguiente                clase 214
```

La fase de auditoría, cuatro semanas.

```
recursos totales                              14.200

incumplimientos por iniciativa
  etiquetas obligatorias                       9.140  64 %
  cifrado en reposo                            1.870  13 %
  sin acceso público                             340   2,4 %
  TLS mínimo                                     620   4,4 %
  regiones permitidas                            112   0,8 %
  tipos de recurso                                88   0,6 %
  sin IP pública                                 210   1,5 %

y los que preocupaban de verdad
  340 recursos con acceso público
    de ellos, cuentas de almacenamiento          214
    de ellas, con datos de clientes               19  ←
  112 recursos fuera de las regiones permitidas
```

```
    de ellos, 8 con datos personales fuera de la UE  ←
```

Y la decisión sobre el orden de remediación:

```
no se atacó lo más numeroso (etiquetas, 9.140)
se atacó lo más grave: los 19 almacenes públicos con datos
y los 8 recursos fuera de región
→ resueltos en 9 días
→ las etiquetas, en 6 semanas con tareas de remediación
```

El hallazgo: mover las suscripciones no arregló casi nada.

```
al mover las 61 suscripciones a su grupo correcto
  políticas heredadas aplicando                    sí
  recursos existentes corregidos                    0

  porque las políticas de denegación solo actúan sobre
  creaciones y modificaciones
  → los 14.200 recursos existentes siguieron exactamente
    igual
  → aparecieron como no conformes, y nada más

lo que sí corrigió recursos existentes
  las políticas de tipo MODIFICAR con tareas de remediación
  → añadieron etiquetas a 8.100 recursos en 3 días
  → y el resto hubo que tocarlo a mano o redesplegarlo

lo que NO se pudo remediar automáticamente
  cifrado en reposo de recursos ya creados: 1.870
  → exige recrear el recurso en la mayoría de los casos
  → 1.410 se recrearon en 4 meses
  →   460 quedaron con exención, con dueño y fecha
```

El paso a denegar, y la lección de la comunicación.

```
primer intento
  se cambiaron 7 iniciativas a denegar el mismo día
  a las 4 horas, 23 despliegues bloqueados
  el mensaje decía «denegado por política»
  → 11 tiquetes de soporte en una tarde

corrección
  se volvió a auditoría
  se añadió a cada política un mensaje con
    qué falta exactamente
    el enlace a la plantilla que ya lo cumple
    y a quién pedir una exención, con el plazo
  y la plantilla de servicio nuevo se actualizó para
    cumplirlas todas                            clase 171

segundo intento, dos semanas después
  despliegues bloqueados en la primera semana         4
  de ellos, incumplimientos reales                    4
  tiquetes de soporte                                 0
  → porque el mensaje decía qué hacer
```

La entrega de suscripciones, automatizada:

```
antes   petición por correo → 3 semanas
        y 12 de las 61 suscripciones se habían creado por
        fuera del proceso                           ley 16

después una plantilla en la canalización crea
          la suscripción en el grupo correcto
          red conectada al centro, con rangos del plan
          identidad y permisos base
          diagnóstico enrutado
          presupuesto y etiquetas
        plazo                                    22 min
        suscripciones creadas por fuera en 6 meses      0
```

El resultado, seis meses después:

```
suscripciones sin grupo de administración      47           0
políticas asignadas en suscripciones            9           0
políticas quitadas por equipos                  4      imposible
```

```
recursos conformes (etiquetas)                36 %        98 %
almacenes públicos con datos                   19           0
recursos fuera de región permitida            112           0
exenciones vivas                                0          22
  con dueño, motivo y caducidad                 —          22
  vencidas y sin renovar                        —           1
plazo de entrega de una suscripción       3 semanas     22 min
suscripciones creadas por fuera del proceso    12           0
```

La lección que esta clase deja: mover sesenta y una suscripciones al grupo correcto no corrigió ni un solo recurso existente, porque las políticas de denegación solo actúan al crear o modificar; lo que corrigió catorce mil recursos fue una combinación de políticas de tipo modificar, tareas de remediación y cuatro meses de recrear cosas. Y las cuatro políticas que los equipos habían quitado no las quitaron por mala fe: las quitaron porque el mensaje de error decía «denegado por política» y no decía qué faltaba.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-18-azure-production-architecture/217-enterprise-scale-landing-zones-y-management-groups/lab.py
```

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

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

Evidencia esperada

El artefacto principal es una jerarquía de política, ownership y evidencia. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye azure-landing-zone 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 equipo quita la política que le estorba y nadie se entera	La política estaba asignada en la suscripción, que ese equipo administra	Asigna siempre en el grupo de administración y alerta sobre cambios de asignación.
Mover las suscripciones no corrige ningún recurso	Las políticas de denegación solo actúan sobre creaciones y modificaciones	Usa políticas de tipo modificar con tareas de remediación para lo existente, y planifica recrear lo que no se pueda corregir.
Al activar el modo denegar se bloquean despliegues legítimos y llueven los tiquetes	Se pasó de cero a denegar sin auditar y sin mensajes útiles	Audita semanas, corrige lo real, añade mensajes que digan qué falta y actualiza la plantilla por defecto antes de denegar.
Una carga impide crecer a otra sin relación con ella	Comparten suscripción y por tanto cuota	Separa por carga y por entorno, y pide cuota con margen antes de necesitarla.
Aparecen recursos creados fuera del proceso	Conseguir una suscripción tarda semanas	Automatiza la entrega de suscripciones preparadas; si tarda minutos, nadie busca atajos.
La factura crece tras aplicar el gobierno	Una política despliega agentes o ajustes de diagnóstico en cada recurso nuevo	Estima el coste de lo que la política despliega antes de asignarla y ajusta categorías y caducidad.

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 resuelve cada uno de los tres niveles de la jerarquía?
2. ¿Por qué las políticas se asignan en grupos de administración y no en suscripciones?
3. ¿Qué corrige mover una suscripción de grupo y qué no?
4. ¿Cuál es el orden correcto para implantar una política y por qué?
5. ¿Qué debe llevar una exención para ser útil?

Referencias

- Microsoft (2025). Cloud Adoption Framework: Azure landing zone design areas. <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/landing-zone/>
- Microsoft (2025). Management groups and organizing resources. <https://learn.microsoft.com/en-us/azure/governance/management-groups/overview>
- Microsoft (2025). Azure Policy effects and remediation. <https://learn.microsoft.com/en-us/azure/governance/policy/concepts/effects>
- Microsoft (2025). Azure Policy exemptions. <https://learn.microsoft.com/en-us/azure/governance/policy/concepts/exemption-structure>
- Microsoft (2025). Subscription vending and landing zone automation. <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/landing-zone/design-area/subscription-vending>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 18 en PDF · Recorrido de Azure en PDF · Manual integral

Anterior	Índice	Siguiente
← 216 · Proyecto: CloudShop productivo en AWS	Parte 18 · Programa	218 · Entra ID, workload identity, PIM y Conditional Access →

Evaluación — 217 Enterprise-scale landing zones y management groups

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 azure-landing-zone 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.

218 — Entra ID, workload identity, PIM y Conditional Access

Parte: 18 — Azure: arquitectura empresarial y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: iam · Estado: EXECUTABLECORE

Propósito

Resolver la identidad en Azure, que es donde se decide el alcance real de todo lo demás. La clase cubre las identidades de carga sin secretos, el modelo de asignación de permisos con su error característico —un ámbito demasiado amplio, que es el equivalente exacto del comodín de la clase 206—, la elevación temporal para el acceso administrativo y el acceso condicional como control de contexto que se aplica a personas y a cargas.

Resultados de aprendizaje

Al finalizar podrás:

1. Eliminar secretos de aplicación usando identidades administradas y federación.
2. Asignar permisos en el ámbito mínimo y detectar los amplios.
3. Configurar elevación temporal con aprobación para el acceso administrativo.
4. Aplicar acceso condicional sin quedarse fuera del propio inquilino.
5. Revisar accesos periódicamente y retirar lo que no se usa.

Conceptos centrales

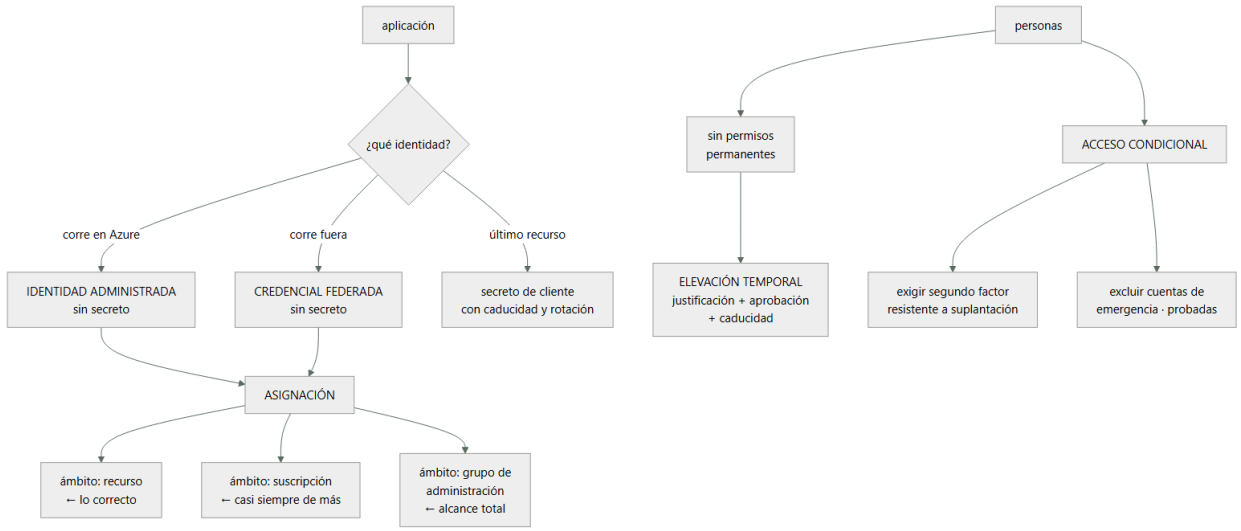
Concepto	Comprensión verificable
identidad administrada	Identidad ligada a un recurso de Azure, sin credenciales que gestionar. Asignada por el sistema o por el usuario.
credencial federada	Confianza declarada en un emisor externo para que sus testigos obtengan credenciales sin secreto.
ámbito de asignación	Nivel al que se concede un permiso: grupo de administración, suscripción, grupo de recursos o recurso.
elevación temporal	Activación de un permiso administrativo por tiempo limitado, con justificación y aprobación.
acceso condicional	Regla que decide si una autenticación se permite según usuario, aplicación, dispositivo, red y riesgo.
acceso de emergencia	Cuentas excluidas de las políticas, para no quedarse fuera. Vigiladas y probadas.

Modelo mental

La arquitectura Azure nace del tenant y las políticas heredables, y llega al workload mediante identidades administradas y redes privadas.

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. Identidad de carga sin secretos

El objetivo es el mismo de la clase 206: que ninguna aplicación tenga una credencial guardada.

```
IDENTIDAD ADMINISTRADA — para lo que corre EN Azure
  el recurso tiene identidad propia; la plataforma le
  entrega testigos
  sin secretos, sin rotación, sin nada que robar

  ASIGNADA POR EL SISTEMA
    nace y muere con el recurso
    + no queda huérfana
    – no se puede preasignar permisos antes de crearlo

  ASIGNADA POR EL USUARIO
    recurso independiente, reutilizable por varios
    + permisos preparados antes; sirve para conjuntos
    – sobrevive al recurso                          ley 25
    → y hay que retirarla cuando ya no la use nadie

CREDENCIAL FEDERADA — para lo que corre FUERA
  canalizaciones, cargas en otro clúster, otra nube
  se declara confianza en el emisor y en el sujeto
  → mismo mecanismo y mismos errores que en la clase 206

SECRETO DE CLIENTE — último recurso
  con caducidad corta, rotación automatizada y una
  excepción registrada con fecha                    ley 25
```

Y la regla que ordena la elección:

```
¿corre en Azure?        identidad administrada
¿corre fuera?           credencial federada
¿ninguna de las dos?    secreto, con fecha de revisión

→ y una función de aptitud: cero secretos de cliente sin
  excepción registrada                          clase 190
```

Y el detalle de configuración donde se falla, igual que en AWS:

```
LA CREDENCIAL FEDERADA SE ATA A
  emisor, sujeto y audiencia

✗ sujeto con comodín, o sin comprobar audiencia
  → cualquier repositorio u organización puede obtener
    el testigo
✓ sujeto exacto: organización, repositorio y entorno

→ es el mismo fallo de la clase 206 con otro nombre, y por
  eso la prueba negativa es idéntica: intentarlo desde otro
```

```
  origen y comprobar que falla                    ley 22
```

2. El ámbito de asignación: el error característico

En Azure los permisos se conceden asignando un papel a una identidad en un ámbito, y el ámbito se hereda hacia abajo. Ahí está el error.

```
LOS CUATRO ÁMBITOS, de mayor a menor
  grupo de administración   → todas sus suscripciones
  suscripción               → todos sus grupos y recursos
  grupo de recursos         → todo lo de dentro
  recurso                   → solo ese

Y LA COSTUMBRE
  asignar «Colaborador» en la suscripción, porque es
  cómodo y funciona
  → esa identidad puede crear, modificar y borrar
    CUALQUIER cosa de la suscripción
  → incluida la base de datos de producción
```

Y el equivalente exacto de la clase 206:

```
en AWS   una condición de sujeto con comodín permite que
         cualquiera asuma el rol
en Azure una asignación en un ámbito amplio permite que la
         identidad toque cualquier cosa de ese ámbito

→ el mecanismo es distinto; el resultado, el mismo:
  alcance enorme desde un punto comprometido    clase 189
```

Cómo se hace bien, con el coste que tiene:

```
asignar en el ÁMBITO DEL RECURSO o del grupo de recursos
con el papel MÁS ESPECÍFICO que exista
  no «Colaborador» sino «Colaborador de datos de blob»
  no «Propietario» nunca, salvo casos contados

y cuando no existe un papel adecuado
  papel personalizado, con las acciones justas
  → cuesta mantenerlo, y compensa en los casos de mucho
    alcance
```

Y los papeles que hay que tratar como excepcionales:

```
Propietario                 puede conceder permisos
                            → quien lo tiene, lo tiene todo
Administrador de acceso     puede conceder permisos
                            → equivalente en la práctica
Colaborador en suscripción  puede borrar casi todo

→ estos tres se conceden con elevación temporal, nunca de
  forma permanente
```

Cómo detectar los ámbitos amplios, que es el trabajo continuo:

```
inventario de asignaciones, por ámbito
  ¿cuántas hay en grupo de administración?
  ¿cuántas en suscripción?
  ¿cuántas identidades tienen Propietario?

y la medida que importa
  ALCANCE desde cada identidad: a cuántos recursos llega
  → se mide, y es la cifra que ordena el trabajo
                                        clases 133, 189

y las asignaciones que no se usan
  → los registros de actividad dicen qué permisos se
    ejercen de verdad
  → lo concedido y no usado se retira            clase 134
```

Y una fuente de alcance que se olvida:

```
LOS GRUPOS ANIDADOS
  un grupo dentro de otro hereda sus asignaciones
  → alguien añadido a un grupo «de proyecto» acaba con
    permisos de producción sin que nadie lo decida
```

```
→ hay que revisar la pertenencia efectiva, no la directa
```

3. Elevación temporal y revisión de accesos

Para las personas, el objetivo es que nadie tenga permisos administrativos permanentes.

```
EL MODELO
  la persona es ELEGIBLE para un papel, no lo tiene
  cuando lo necesita, lo activa
    con justificación escrita
    con aprobación de otra persona, si el papel es
      sensible
    con segundo factor en el momento de activar
    y con CADUCIDAD: 1, 4 u 8 horas

QUÉ RESUELVE
  el permiso permanente que nadie revisa
  el administrador que se va y conserva accesos
  el compromiso de una cuenta con permisos totales
  y deja registro de POR QUÉ se usó, no solo de que se usó
                                                clase 141
```

Y los detalles que hacen que funcione o que se rodee:

```
SI ACTIVAR TARDA MUCHO, la gente pedirá permanente
  → activación en segundos cuando no requiere aprobación
  → y aprobadores suficientes, con turnos, cuando sí
                                                  ley 16

SI NO HAY APROBADOR DISPONIBLE DE MADRUGADA
  → un incidente se atasca esperando
  → por eso los papeles de emergencia se activan sin
    aprobación, con alerta inmediata y revisión posterior

Y LA PRUEBA QUE HAY QUE HACER
  activar el papel de emergencia y cronometrar
  → en la clase 179, el acceso de emergencia estaba creado,
    documentado y NO funcionaba                   ley 22
```

Las revisiones de acceso, que es lo que impide la acumulación:

```
cada trimestre, quien es dueño revisa
  quién es elegible para cada papel
  qué identidades tienen asignaciones amplias
  qué invitados externos siguen teniendo acceso

y la regla que las hace útiles
  quien no responde a la revisión → se retira el acceso
  → si la falta de respuesta lo conserva, la revisión no
    sirve de nada
```

Y la señal que dice si el modelo está funcionando:

```
número de asignaciones PERMANENTES de papeles sensibles
  objetivo: cero, salvo cuentas de emergencia
y número de activaciones al mes, con su justificación
  → si nadie activa nunca, o sobran los papeles o la gente
    tiene permisos por otra vía
```

4. Acceso condicional, sin quedarse fuera

El acceso condicional decide si una autenticación se permite según el contexto, y es el control más potente del directorio.

```
LO QUE PUEDE EVALUAR
  quién: usuario, grupo, papel
  qué: aplicación o recurso
  desde dónde: red, país, dispositivo
  con qué: dispositivo conforme o unido al directorio
  y el riesgo: de la sesión y del usuario

LO QUE PUEDE EXIGIR
  segundo factor
  dispositivo conforme
  aplicación cliente aprobada
```

```
  frecuencia de reautenticación
  o directamente bloquear
```

Las políticas mínimas de una organización:

```
1  segundo factor para TODOS los usuarios
   → y resistente a suplantación (llave física o similar)
     para los papeles administrativos
2  bloquear protocolos de autenticación antiguos
   → son los que se saltan el segundo factor
3  exigir dispositivo conforme para el acceso
   administrativo
4  bloquear países donde no se opera
   → medida sencilla que reduce mucho el ruido
5  exigir reautenticación al elevar privilegios
6  y proteger los flujos de registro de credenciales
   → registrar un segundo factor es un momento crítico
```

Y el error que deja a la organización fuera de su propio inquilino:

```
UNA POLÍTICA QUE EXIGE DISPOSITIVO CONFORME A TODOS
  incluidas las cuentas administrativas
  y el día que falla el sistema de conformidad, o el
  dispositivo del administrador se rompe
  → nadie puede entrar a arreglarlo

LA PROTECCIÓN
  cuentas de acceso de emergencia
    excluidas de todas las políticas condicionales
    con credenciales largas guardadas físicamente
    con alerta INMEDIATA en cada uso
    y probadas cada trimestre                    ley 22
  y siempre DOS, por si una falla
```

Y la disciplina de despliegue, que es la misma de siempre:

```
1  crear la política en modo SOLO INFORME
2  medir a quién afectaría, durante semanas
3  excluir lo que haya que excluir, con registro
4  activar para un grupo piloto
5  activar para todos
→ activar una política condicional para todos el primer día
  es la forma más rápida de dejar a la empresa sin acceso
                                              clase 200
```

Y la lista de comprobación de la clase:

- ninguna aplicación tiene secretos de cliente sin excepción registrada
- lo que corre en Azure usa identidad administrada
- las credenciales federadas atan emisor, sujeto y audiencia
- no hay asignaciones de Propietario permanentes
- las asignaciones se hacen en el ámbito del recurso o del grupo
- se usa el papel más específico disponible
- está medido el alcance desde cada identidad
- se revisan los permisos concedidos y no usados
- se revisa la pertenencia efectiva a grupos, no la directa
- los papeles sensibles requieren elevación temporal
- hay aprobadores suficientes y activación rápida
- el acceso de emergencia se ha probado este trimestre
- hay dos cuentas de emergencia excluidas de las políticas
- cada uso de emergencia dispara alerta inmediata
- las políticas condicionales se desplegaron en solo informe
- los protocolos de autenticación antiguos están bloqueados

Y el cierre que enlaza con la clase siguiente: con la jerarquía y la identidad resueltas, queda conectar las suscripciones entre sí y con el centro de datos, y decidir por dónde entra y sale el tráfico. Red en Azure es la materia de la clase 219.

Ejemplo trabajado

CloudShop revisa la identidad de su inquilino de Azure. Lo que sigue es el inventario de asignaciones, las tres identidades que podían tocar toda la producción, y la política condicional que dejó al equipo fuera durante cuarenta minutos.

El inventario:

```
identidades de aplicación                          184
  con identidad administrada                        61
  con credencial federada                            9
  CON SECRETO DE CLIENTE                           114  ←
    de ellos, caducados y aún referenciados         18
    de ellos, con más de 2 años                     47

asignaciones de papel                            1.940
  en grupo de administración                        31  ←
  en suscripción                                   820  ←
  en grupo de recursos                             760
  en recurso                                       329

papeles sensibles, permanentes
  Propietario                                       41
  Administrador de acceso                           12
  Colaborador en suscripción                       214
```

Y las tres asignaciones que resultaron peores:

```
1  la identidad de la canalización de despliegue tenía
   Colaborador en el grupo de administración «cargas»
   → alcance: 43 suscripciones, todo
   → y podía borrar bases de datos de producción

2  una identidad administrada asignada por el usuario,
   creada en 2022 para una prueba, tenía Propietario en la
   suscripción de producción de pedidos
   → el recurso que la usaba se borró hace 2 años
   → la identidad seguía viva, con Propietario     ley 25

3  un grupo llamado «Proyecto Migración 2023» tenía
   Colaborador en 9 suscripciones
   → 34 personas seguían en él
   → 11 ya no trabajaban en esas cargas
   → y 3 habían entrado por pertenencia ANIDADA, sin que
     nadie los añadiera a propósito
```

El alcance medido, que es lo que ordenó el trabajo:

```
identidad                     recursos alcanzables
canalización de despliegue              14.200   ← todo
identidad huérfana de 2022               1.840
grupo Proyecto Migración                 6.100
api-pedidos (correcta)                       4
procesador-eventos (correcta)                7

→ dos identidades y un grupo concentraban casi todo el
  riesgo del inquilino
```

Las correcciones:

```
CANALIZACIÓN
  antes    Colaborador en grupo de administración
  después  4 identidades federadas, una por propósito
           cada una con asignación en el GRUPO DE RECURSOS
           que le toca
           y una barrera: ninguna puede asignar papeles ni
           modificar políticas
  alcance  14.200 → 340 recursos, repartidos en 4

IDENTIDAD HUÉRFANA
  retirada
  y una comprobación nueva: identidades administradas
  asignadas por el usuario sin ningún recurso asociado
  → se encontraron 23 más                          ley 25

GRUPO DE PROYECTO
  retirado; sustituido por grupos por carga y entorno
  y revisión trimestral de pertenencia EFECTIVA

SECRETOS DE CLIENTE
  114 → 12 en cuatro meses
```

```
    61 pasaron a identidad administrada
    28 a credencial federada
    13 se apagaron: eran aplicaciones que ya no existían
    12 quedan, todos de terceros, con rotación automática
       y excepción registrada con fecha
  los 18 caducados y referenciados
    → 4 de ellos, en aplicaciones que fallaban
      silenciosamente desde hacía meses            ley 13
```

La elevación temporal:

```
se retiraron TODAS las asignaciones permanentes de
Propietario y Administrador de acceso
y pasaron a elegibles con activación

  Propietario                aprobación de 2, caducidad 4 h
  Administrador de acceso    aprobación de 2, caducidad 2 h
  Colaborador en suscripción activación directa, 8 h
  Emergencia                 sin aprobación, alerta
                             inmediata, revisión posterior

los primeros dos meses
  activaciones                                      412
  con justificación útil                            340
  con justificación tipo «trabajo»                   72
    → se pidió mejorarlas; a los 3 meses, 11
  activaciones de emergencia                          3
    · 2 incidentes reales
    · 1 porque el aprobador estaba de vacaciones y nadie
      había cubierto el turno            ← se corrigió

y la prueba de emergencia, ejecutada
  primer intento   la cuenta de emergencia no podía activar
                   porque la política condicional creada en
                   el paso siguiente la había incluido
                   sin querer
  → corregido, y añadido al calendario trimestral
```

La política condicional que dejó al equipo fuera.

```
se creó una política: «exigir dispositivo conforme para
acceder al portal de Azure»
se activó directamente para todos

  a las 09:40, el sistema de evaluación de conformidad
  tuvo una incidencia y marcó como no conformes a 190
  dispositivos
  → nadie del equipo de plataforma podía entrar al portal
  → incluidas las 2 cuentas de emergencia, que se habían
    incluido en el ámbito por error

  09:40  se detecta
  09:52  se localiza a alguien con acceso desde otro camino
  10:20  política desactivada
  duración                                        40 min

qué faltó
  modo solo informe durante semanas
  exclusión explícita y COMPROBADA de las cuentas de
    emergencia
  y despliegue por grupos                    clase 200

segundo intento
  1  solo informe, 3 semanas
     → 1.140 accesos que se habrían bloqueado
     → de ellos, 210 legítimos desde dispositivos no
       registrados: contratistas y 2 equipos con portátiles
       propios
  2  se registraron los dispositivos legítimos
  3  exclusión de cuentas de emergencia, comprobada
     entrando con ellas
  4  grupo piloto de 12 personas, 2 semanas
  5  activación general
  accesos bloqueados legítimos tras activar          2
```

El resto de políticas, con su despliegue:

```
política                              informe   activa
segundo factor para todos               3 sem     sí
segundo factor resistente para
  papeles administrativos               2 sem     sí
bloquear autenticación antigua          4 sem     sí
  → el informe encontró 6 aplicaciones que la usaban
  → 4 se corrigieron, 2 se apagaron
dispositivo conforme para portal        3 sem     sí
bloquear países no operados             1 sem     sí
  → 41.000 intentos bloqueados el primer mes
reautenticación al elevar               1 sem     sí
```

El resultado, seis meses después:

```
secretos de cliente                          114          12
  caducados y referenciados                   18           0
asignaciones en grupo de administración       31           4
asignaciones en suscripción                  820         190
Propietario permanente                        41           0
alcance de la identidad de despliegue     14.200         340
identidades huérfanas                         24           0
personas con permisos administrativos
  permanentes                                267           2
  (las 2 cuentas de emergencia)
prueba de emergencia ejecutada                 no    trimestral
incidentes por política condicional              1           0
```

La lección que esta clase deja: el error característico de Azure resultó ser exactamente el equivalente del comodín de la clase 206: no una condición mal escrita, sino un ámbito de asignación demasiado alto, con una sola identidad capaz de tocar catorce mil recursos. Y la política de seguridad que se activó para proteger el portal dejó a la empresa fuera durante cuarenta minutos, incluidas las cuentas de emergencia que existían precisamente para eso y que nadie había comprobado.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-18-azure-production-architecture/218-entra-id-workload-identity-pim-y-conditional-access/lab.py
```

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

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

Evidencia esperada

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

Reto verificable

Construye azure-identity 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 identidad de aplicación puede tocar recursos que no tienen nada que ver con ella	La asignación se hizo en la suscripción o en el grupo de administración	Asigna en el ámbito del recurso o del grupo de recursos, con el papel más específico disponible, y mide el alcance resultante.
Alguien acaba con permisos de producción sin que nadie se los diera	Pertenencia anidada a un grupo con asignaciones amplias	Revisa la pertenencia efectiva, no la directa, y retira los grupos de proyecto cuando el proyecto termina.
Una identidad sigue teniendo permisos y su recurso ya no existe	Identidad administrada asignada por el usuario, que sobrevive al recurso	Inventaría periódicamente las identidades sin recurso asociado y retíralas.
Una aplicación falla en silencio desde hace meses	Su secreto de cliente caducó y nadie lo vigilaba	Sustituye por identidad administrada o federada; para lo que quede, rotación automática y alerta por caducidad próxima.
Nadie puede entrar al portal cuando falla un sistema auxiliar	Una política condicional se activó para todos sin exclusiones comprobadas	Despliega en solo informe, excluye dos cuentas de emergencia y comprueba entrando con ellas antes de activar.
La gente pide permisos permanentes pese a existir elevación temporal	Activar tarda demasiado o no hay aprobadores disponibles	Activación en segundos cuando no requiere aprobación, aprobadores con turnos y un papel de emergencia sin aprobación con alerta inmediata.

■ 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 identidad corresponde a una carga que corre en Azure y cuál a una que corre fuera?
2. ¿Cuál es el equivalente en Azure del comodín en la condición de sujeto de AWS?
3. ¿Qué hace que la elevación temporal se rodee y cómo se evita?
4. ¿Por qué hacen falta dos cuentas de acceso de emergencia y qué hay que probar de ellas?
5. ¿Cómo se despliega una política condicional sin dejar a la organización fuera?

Referencias

- Microsoft (2025). Managed identities for Azure resources. <https://learn.microsoft.com/en-us/entra/identity/managed-identities-azure-resources/overview>
- Microsoft (2025). Workload identity federation. <https://learn.microsoft.com/en-us/entra/workload-id/workload-identity-federation>
- Microsoft (2025). Azure RBAC scope and best practices. <https://learn.microsoft.com/en-us/azure/role-based-access-control/best-practices>
- Microsoft (2025). Privileged Identity Management. <https://learn.microsoft.com/en-us/entra/id-governance/privileged-identity-management/pim-configure>

- Microsoft (2025). Conditional Access and emergency access accounts. <https://learn.microsoft.com/en-us/entra/identity/role-based-access-control/security-emergency-access>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 18 en PDF · Recorrido de Azure en PDF · Manual integral

Anterior	Índice	Siguiente
← 217 · Enterprise-scale landing zones y management groups	Parte 18 · Programa	219 · Hub-spoke, Virtual WAN, Private Link y DNS privado →

Evaluación — 218 Entra ID, workload identity, PIM y Conditional Access

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 azure-identity 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.

219 — Hub-spoke, Virtual WAN, Private Link y DNS privado

Parte: 18 — Azure: arquitectura empresarial y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: network · Estado: EXECUTABLECORE

Propósito

Montar la red de Azure conectando decenas de suscripciones sin abrir caminos indeseados. La clase cubre la topología de centro y radios frente a la gestionada, los puntos privados hacia los servicios de la plataforma, y la parte que produce la mayoría de los incidentes de esta capa: la resolución de nombres privados, que en Azure exige enlazar cada zona a cada red y falla en silencio cuando no se hace.

Resultados de aprendizaje

Al finalizar podrás:

1. Elegir entre centro y radios propio y topología gestionada.
2. Segmentar con grupos de seguridad y rutas, cerrando por ausencia de camino.
3. Conectar a los servicios de la plataforma con puntos privados.
4. Resolver nombres privados en todas las redes, sin huecos.
5. Diagnosticar los fallos característicos de esta topología.

Conceptos centrales

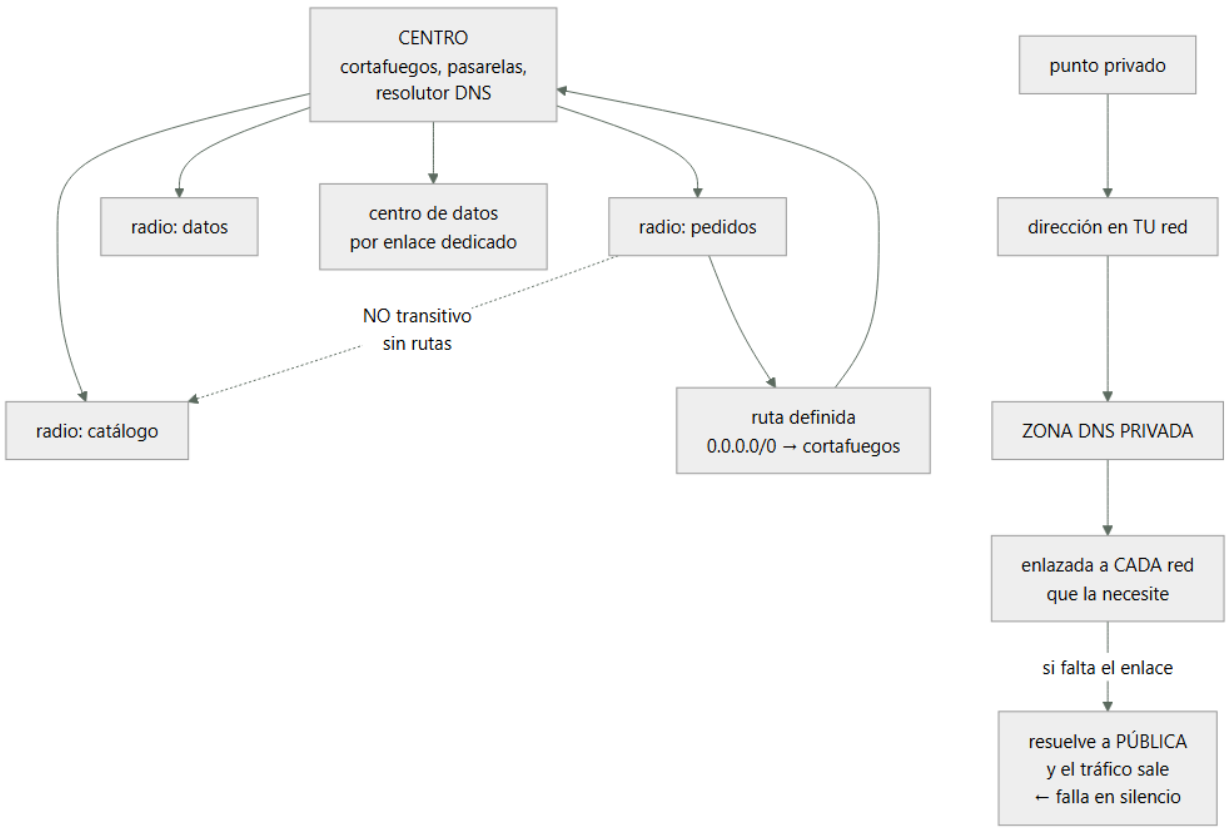
Concepto	Comprensión verificable
centro y radios	Topología con una red central que concentra conectividad e inspección, y redes de carga conectadas a ella.
emparejamiento	Conexión entre dos redes virtuales. No es transitivo por sí solo: hace falta encaminar.
ruta definida por el usuario	Ruta que obliga al tráfico a pasar por un dispositivo. Es lo que fuerza la inspección.
punto privado	Interfaz con dirección de tu red que representa un servicio de la plataforma.
zona DNS privada	Zona que resuelve el nombre del servicio a la dirección privada. Debe enlazarse a cada red que la necesite.
grupo de seguridad de red	Reglas de filtrado por subred o interfaz, con prioridades y reglas por defecto que hay que conocer.

Modelo mental

La arquitectura Azure nace del tenant y las políticas heredables, y llega al workload mediante identidades administradas y redes privadas.

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. Topología: centro y radios

La topología estándar de Azure concentra en una red central lo que no debe repetirse.

```
EN EL CENTRO
  cortafuegos o dispositivo de inspección
  pasarelas hacia el centro de datos y hacia internet
  resolutor DNS y sus reglas de reenvío
  y, a veces, los puntos privados compartidos

EN CADA RADIO
  las cargas, en sus subredes
  sin pasarela propia ni salida directa a internet
```

Y la propiedad que hay que entender antes de nada:

```
EL EMPAREJAMIENTO NO ES TRANSITIVO
  radio A ↔ centro y radio B ↔ centro
  → A NO alcanza B automáticamente
  → hace falta que el tráfico pase por un dispositivo del
    centro y haya rutas que lo lleven allí

→ y esto es una VENTAJA: el aislamiento entre radios es el
  estado por defecto                          clase 199
→ abrirlo es una decisión, no un descuido
```

La elección entre montar el centro o usar la topología gestionada:

```
CENTRO PROPIO
  + control total: cualquier dispositivo, cualquier diseño
  + más barato en tráfico
  – hay que montar y operar el encaminamiento y la
    alta disponibilidad

TOPOLOGÍA GESTIONADA (Virtual WAN)
  + conecta radios, sedes y usuarios remotos con menos
    trabajo
  + encaminamiento y escala gestionados
```

```
  + tablas de rutas para segmentar              clase 199
  − coste por conexión y por gigabyte
  − menos control fino

CRITERIO
  pocas redes y una región    → centro propio
  muchas sedes, varias regiones, usuarios remotos
                              → gestionada
```

Y lo que hay que decidir en ambos casos:

```
¿POR DÓNDE SALE EL TRÁFICO A INTERNET?
  salida centralizada por el cortafuegos del centro
  → registro, control y direcciones fijas       clase 200
  → y una ruta definida por el usuario en cada subred que
    mande 0.0.0.0/0 al cortafuegos

¿QUÉ RADIOS SE VEN ENTRE SÍ?
  por defecto, ninguno
  → y lo que se abra, con rutas explícitas y pasando por
    inspección si cruza fronteras de confianza  clase 189
```

Y una decisión que ahorra dinero y latencia:

```
el tráfico entre dos radios que se hablan mucho puede ir
directo con un emparejamiento entre ellos
→ evita pagar el proceso del cortafuegos dos veces
→ y hay que decidirlo, no dejarlo pasar por costumbre
                                                clase 199
```

2. Filtrado y rutas: lo que hay que saber

Los grupos de seguridad de red tienen reglas por defecto que sorprenden si no se conocen.

```
REGLAS POR DEFECTO, que existen aunque no se vean
  entrante   permitido desde la propia red virtual
             permitido desde el balanceador
             DENEGADO todo lo demás
  saliente   permitido hacia la propia red virtual
             PERMITIDO HACIA INTERNET            ←
             denegado el resto

→ la salida a internet está PERMITIDA por defecto
→ y por eso el control de salida es una decisión explícita
                                        clase 200, ley 26
```

Y el funcionamiento por prioridad, que causa confusión:

```
las reglas se evalúan por número de prioridad, de menor a
mayor, y la PRIMERA que coincide decide
→ una regla permisiva con prioridad baja anula las
  restrictivas posteriores
→ y las reglas por defecto tienen prioridades muy altas:
  siempre se evalúan al final

regla práctica
  deja huecos entre prioridades (100, 200, 300…)
  y documenta el orden esperado
```

Y dónde aplicarlos:

```
A LA SUBRED     lo habitual; se aplica a todo lo de dentro
A LA INTERFAZ   para excepciones; complica el diagnóstico
→ aplicar en los dos sitios a la vez es la causa más común
  de «no entiendo por qué no llega»
```

Las rutas definidas por el usuario, que son lo que fuerza la inspección:

```
por defecto, cada subred tiene rutas del sistema hacia
  la propia red, los emparejamientos e internet

una ruta definida por el usuario las sustituye
  0.0.0.0/0 → dirección del cortafuegos
  → todo el tráfico saliente pasa por él

y el error clásico
  poner la ruta en la subred del cortafuegos también
```

```
  → el tráfico entra en bucle
  → la subred del dispositivo NO lleva esa ruta
```

Y la regla del prefijo más específico, que aquí también manda:

```
si hay una ruta más específica hacia otro sitio, gana
  → así se implementan las excepciones            clase 194
→ y así también se abren caminos sin querer, si alguien
  añade una ruta «temporal»                        ley 25
```

Y una comprobación que resuelve la mitad de los casos:

```
la herramienta de diagnóstico de siguiente salto dice, para
un origen y un destino concretos, qué ruta se aplica y hacia
dónde va
→ y hay otra que dice qué regla de seguridad permite o
  deniega ese flujo
→ usarlas antes de suponer                       clase 202
```

3. Puntos privados y la trampa del DNS

Los puntos privados en Azure resuelven lo mismo que en la clase 200, y tienen una particularidad que causa la mayoría de los incidentes de esta clase.

```
CÓMO FUNCIONA
  se crea un punto privado en una subred: obtiene una
  dirección de TU red
  el servicio queda alcanzable por esa dirección
  y se puede desactivar su acceso público

Y LA PARTE QUE FALLA
  el nombre público del servicio debe resolver a la
  dirección PRIVADA
  eso lo hace una ZONA DNS PRIVADA
  y esa zona debe estar ENLAZADA a cada red virtual que la
  necesite

  ✗ si falta el enlace
    el nombre resuelve a la dirección PÚBLICA
    → el tráfico sale a internet
    → y si el acceso público está desactivado, FALLA
    → y si no lo está, FUNCIONA pagando salida y sin
      pasar por ningún control            ← lo peor
```

Y por eso el fallo es silencioso:

```
funciona desde la red donde se creó el punto privado
y no funciona —o funciona por internet— desde las demás
→ el síntoma es «a mí me va bien»                clase 195
```

El montaje que evita el problema:

```
ZONAS PRIVADAS CENTRALIZADAS
  todas las zonas de puntos privados en la suscripción de
  conectividad
  enlazadas a TODAS las redes que las necesitan
  y creadas automáticamente por política          clase 217
    → una política de tipo «desplegar si no existe» que
      registra el punto privado en la zona correcta
    → sin ella, cada equipo crea su zona y aparecen
      duplicadas y divergentes

RESOLUTOR PRIVADO EN EL CENTRO
  con regla de reenvío desde el centro de datos hacia
  Azure y viceversa
  → y hay que comprobarlo EN LOS DOS SENTIDOS  clase 195

COMPROBACIÓN AUTOMÁTICA
  para cada servicio con punto privado, resolver su nombre
  desde CADA red y verificar que devuelve dirección privada
  → función de aptitud, ejecutada a diario     clase 190
```

Y una decisión que hay que tomar y que se olvida:

```
DESACTIVAR EL ACCESO PÚBLICO del servicio
  crear el punto privado no cierra el público
```

```
  → el servicio sigue alcanzable desde internet
  → y la política de «sin acceso público» de la clase 217
    es lo que lo impide de verdad
```

Y el consumo de direcciones, que hay que planificar:

```
un punto privado = una dirección de la subred
30 servicios × 3 entornos = 90 direcciones
→ y a eso se suman las subredes que ciertos servicios
  exigen en exclusiva y con tamaño mínimo
→ el plan de direcciones tiene que contarlo    clase 193
```

4. Diagnóstico y operación

Los fallos de esta topología tienen firmas reconocibles.

SÍNTOMA	CAUSA HABITUAL
funciona desde una red y no desde otra	falta el enlace de la zona DNS privada
el tráfico sale a internet pese a tener punto privado	el nombre resuelve a la dirección pública
dos radios no se ven	el emparejamiento no es transitivo y faltan rutas
el tráfico no pasa por el cortafuegos	falta la ruta definida por el usuario en esa subred
bucle de encaminamiento	la ruta 0.0.0.0/0 se puso también en la subred del cortafuegos
se permite algo que no debería	una regla permisiva con prioridad menor gana
no llega y las reglas parecen correctas	hay grupo de seguridad en la subred Y en la interfaz
todo deja de funcionar tras un cambio de red	alguien cambió una ruta y el prefijo más específico desvió el tráfico

Y las herramientas, por orden de coste:

```
1  comprobación de siguiente salto y de flujo permitido
   → responde en segundos qué ruta y qué regla aplican
2  registros de flujo de los grupos de seguridad
   → quién habla con quién, qué se deniega   clase 202
3  captura de paquetes
   → solo cuando lo anterior no basta
```

Lo que hay que vigilar:

```
zonas privadas sin enlazar a redes que las necesitan
puntos privados cuyo nombre resuelve a dirección pública
reglas de seguridad con origen o destino demasiado amplios
rutas definidas por el usuario creadas fuera del proceso
estado de las pasarelas y de las sesiones hacia el centro
  de datos                                    clase 198
uso de direcciones por subred, frente al total
  → una subred agotada bloquea el escalado     clase 193
```

Y una alerta que evita sorpresas:

```
«ha aparecido una ruta o una regla que no está en el
 repositorio»
→ detecta los cambios manuales, que son los que se quedan
                                                  ley 25
```

Y la lista de comprobación de la clase:

- ■ la topología está decidida y justificada
- ■ los radios no se ven entre sí salvo decisión explícita
- ■ toda subred de carga tiene ruta 0.0.0.0/0 al cortafuegos

- la subred del cortafuegos NO tiene esa ruta
- los grupos de seguridad están en la subred, no en ambos sitios
- las prioridades dejan huecos y están documentadas
- se sabe que la salida a internet está permitida por defecto
- los servicios de plataforma se alcanzan por punto privado
- el acceso público de esos servicios está desactivado
- las zonas DNS privadas están centralizadas y enlazadas a todas las redes
- hay política que registra los puntos privados en la zona correcta
- hay comprobación diaria de que cada nombre resuelve a dirección privada desde cada red
- el reenvío de nombres con el centro de datos funciona en ambos sentidos
- el plan de direcciones cuenta los puntos privados y las subredes exclusivas
- hay alerta ante rutas o reglas creadas fuera del proceso

Y el cierre que enlaza con la clase siguiente: con jerarquía, identidad y red resueltas, hace falta declarar todo esto como código de forma que se pueda desplegar, revisar y retirar. Es la materia de la clase 220.

Ejemplo trabajado

CloudShop monta la red de sus 61 suscripciones en Azure. Lo que sigue es el incidente del punto privado que llevaba meses saliendo a internet, el bucle de encaminamiento del primer día, y la comprobación diaria que quedó montada.

La topología elegida:

```
centro propio, uno por región (2)
  cortafuegos con inspección
  pasarela hacia el centro de datos              clase 198
  resolutor DNS privado con reglas de reenvío
  zonas DNS privadas de todos los servicios

43 radios: uno por carga y entorno
  sin pasarela propia
  sin salida directa a internet
  ruta 0.0.0.0/0 → cortafuegos del centro

motivo de centro propio y no gestionado
  2 regiones, sin sedes remotas ni usuarios remotos
  y el volumen entre radios y centro justificaba el ahorro
  → decisión registrada, con la señal que la reabriría:
    «si se conectan más de 10 sedes»            clase 190
```

Incidente 1 · El bucle de encaminamiento, día uno.

```
síntoma   al aplicar la ruta 0.0.0.0/0 en todas las subredes
          del centro y de los radios, nada funcionaba
          el cortafuegos mostraba tráfico que entraba y
          volvía a entrar

causa     la ruta se había aplicado también a la subred del
          propio cortafuegos
          → el tráfico salía del cortafuegos, encontraba la
            ruta que apuntaba al cortafuegos, y volvía

corrección
  la subred del cortafuegos usa las rutas del sistema
  y se añadió una función de aptitud: ninguna tabla de
  rutas con 0.0.0.0/0 puede asociarse a la subred del
  dispositivo de inspección                     clase 190

tiempo perdido                                    3 h
```

Incidente 2 · El punto privado que salía a internet.

```
síntoma reportado   ninguno

se descubrió al revisar el coste de salida     clase 214
  transferencia de salida a internet          1.940 €/mes
  esperada, con puntos privados                 ~200 €
```

```
diagnóstico
  27 servicios tenían punto privado creado
  las zonas DNS privadas estaban enlazadas a la red del
  centro y a 4 radios
  → y había 43 radios

  desde los 39 radios sin enlace
    el nombre del servicio resolvía a la dirección PÚBLICA
    el tráfico salía por el cortafuegos a internet
    y volvía a entrar por el punto público del servicio
    → funcionaba perfectamente
    → y por eso nadie lo reportó                    ley 13

  cuánto llevaba así
    los primeros radios se crearon hacía 7 meses
    los enlaces se hicieron a mano para los 4 primeros y
    luego nadie los hizo más                       ley 25

lo que implicaba, además del coste
  el tráfico hacia las bases de datos salía a internet
  cifrado, pero por internet
  y el acceso público de esos servicios seguía activado
  → cualquiera con la credencial podía llegar desde fuera
                                                clase 189

corrección
  1  política de tipo «desplegar si no existe» que registra
     cada punto privado en la zona centralizada correcta
  2  enlace de todas las zonas a las 43 redes, automatizado
  3  desactivación del acceso público de los 27 servicios
     → y aquí fallaron 3: dos bases y un almacén tenían
       clientes externos legítimos que se descubrieron al
       cortar
     → 2 se movieron a punto privado desde el centro de
       datos; 1 quedó con excepción, con dueño y fecha
  4  comprobación diaria: resolver el nombre de cada
     servicio desde cada red y verificar dirección privada

resultado
  transferencia de salida             1.940 € → 180 €/mes
  servicios con acceso público            27 → 1 (con
                                          excepción)
  fallos de la comprobación diaria en 6 meses         4
    → los 4, radios nuevos creados antes de que la
      automatización enlazara sus zonas
    → corregidos en menos de 1 hora cada uno
```

Incidente 3 · «No llega y las reglas parecen correctas».

```
síntoma   el servicio de catálogo no podía llamar al de
          precios, en otro radio

lo que se revisó primero (mal)
  las reglas del grupo de seguridad: correctas
  el emparejamiento: existía
  2 horas de revisión manual

lo que lo resolvió, en 90 segundos
  la comprobación de siguiente salto
  → dijo que el tráfico iba al cortafuegos y ahí se
    detenía
  la comprobación de flujo permitido
  → dijo que la regla del CORTAFUEGOS, no la del grupo de
    seguridad, lo denegaba

causa
  el emparejamiento entre radios no es transitivo
  el tráfico pasaba por el centro, correctamente
  y el cortafuegos no tenía regla para ese par
  → se había asumido que el emparejamiento bastaba

corrección
  regla en el cortafuegos, y documentación del camino
```

```
  esperado para los 8 flujos entre radios     clase 194
```

El plan de direcciones, ajustado tras contar:

```
cálculo inicial por radio           /24 (256)

cálculo real
  cargas                                          ~60
  puntos privados (27 servicios)                   27
  subredes exclusivas de servicios gestionados
    pasarela de aplicación (mínimo /24 propia)
    bases gestionadas con integración de red        2 × /26
    entorno de contenedores                        1 × /23
  margen de despliegue escalonado                 ×2
  ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  → /21 por radio, no /24

→ el plan de la clase 193 tuvo que revisarse para Azure
→ y los 4 radios creados con /24 hubo que ampliarlos
  antes de que se agotaran
```

La vigilancia montada:

```
zonas privadas sin enlazar a una red existente   → alerta
nombre de servicio que resuelve a pública        → alerta
reglas de seguridad con origen «cualquiera»      → auditada
rutas creadas fuera del repositorio              → alerta
  → 6 en el primer trimestre; 5 eran temporales de
    diagnóstico que nadie quitó                    ley 25
direcciones libres por subred, alerta al 80 %
estado de la sesión hacia el centro de datos   clase 198
```

El resultado:

```
transferencia de salida                   1.940 €       180 €
servicios con acceso público                   27           1
radios con zonas DNS enlazadas               4/43       43/43
tiempo de diagnóstico de red               2 h 10       9 min
rutas manuales sin registrar                    6           0
subredes agotadas o en riesgo                   4           0
bucles de encaminamiento                        1           0
```

La lección que esta clase deja: los puntos privados estaban creados, se pagaban y el tráfico salía a internet igualmente porque las zonas DNS privadas solo estaban enlazadas a cuatro de cuarenta y tres redes. No falló nada, no saltó ninguna alerta y nadie lo reportó: se descubrió mirando la factura de salida. Y el diagnóstico que costó dos horas revisando reglas a mano se resolvía en noventa segundos con dos comprobaciones que la plataforma ya ofrecía.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-18-azure-production-architecture/219-hub-spoke-virtual-wan-private-link-y-dns-privado/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 azure-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 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 azure-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
Un punto privado existe y el tráfico sigue saliendo a internet	La zona DNS privada no está enlazada a esa red virtual y el nombre resuelve a la dirección pública	Centraliza las zonas, enlázalas a todas las redes por automatización y comprueba a diario que cada nombre resuelve a dirección privada desde cada red.
El servicio sigue siendo alcanzable desde internet pese al punto privado	Crear el punto privado no desactiva el acceso público	Desactiva el acceso público explícitamente y respáldalo con una política de denegación heredada.
Nada funciona tras aplicar la ruta hacia el cortafuegos	La ruta 0.0.0.0/0 se aplicó también a la subred del propio dispositivo	Deja esa subred con las rutas del sistema y comprueba la asociación con una función de aptitud.
Dos radios no se comunican aunque estén emparejados con el centro	El emparejamiento no es transitivo y falta la regla en el cortafuegos	Documenta el camino esperado de cada flujo entre radios y añade la regla correspondiente.
Una regla de seguridad no surte efecto	Otra regla permisiva con prioridad menor decide antes, o hay grupos aplicados en subred y en interfaz	Aplica en la subred, deja huecos entre prioridades y usa la comprobación de flujo permitido antes de suponer.
Una subred se queda sin direcciones y bloquea el escalado	El dimensionado no contó puntos privados ni las subredes exclusivas que exigen ciertos servicios	Cuenta todos los consumidores y dimensiona con margen, revisando el plan para las particularidades de esta nube.

■ 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 aislamiento entre radios es el estado por defecto y qué hace falta para abrirlo?
2. ¿Qué regla por defecto de los grupos de seguridad sorprende y qué implica?
3. ¿Qué ocurre exactamente cuando falta el enlace de una zona DNS privada?
4. ¿Por qué la subred del cortafuegos no debe llevar la ruta hacia el cortafuegos?
5. ¿Qué dos comprobaciones resuelven la mayoría de los diagnósticos de red en Azure?

Referencias

- Microsoft (2025). Hub-spoke network topology in Azure. <https://learn.microsoft.com/en-us/azure/architecture/networking/architecture/hub-spoke>
- Microsoft (2025). Azure Private Link and private endpoint DNS configuration. <https://learn.microsoft.com/en-us/azure/private-link/private-endpoint-dns>
- Microsoft (2025). Network security groups: default rules and evaluation. <https://learn.microsoft.com/en-us/azure/virtual-network/network-security-groups-overview>
- Microsoft (2025). User-defined routes and forced tunneling. <https://learn.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview>
- Microsoft (2025). Azure Virtual WAN architecture. <https://learn.microsoft.com/en-us/azure/virtual-wan/virtual-wan-about>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 18 en PDF · Recorrido de Azure en PDF · Manual integral

Anterior	Índice	Siguiente
← 218 · Entra ID, workload identity, PIM y Conditional Access	Parte 18 · Programa	220 · Bicep, deployment stacks y Azure Verified Modules →

Evaluación — 219 Hub-spoke, Virtual WAN, Private Link y DNS privado

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 azure-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.

220 — Bicep, deployment stacks y Azure Verified Modules

Parte: 18 — Azure: arquitectura empresarial y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: iac · Estado: EXECUTABLECORE

Propósito

Declarar la infraestructura de Azure como código de forma que se pueda desplegar, revisar y —lo que casi nunca se resuelve— retirar. La clase cubre Bicep frente a las plantillas nativas y a las herramientas de terceros, las pilas de despliegue que sí saben borrar lo que sobra, los módulos verificados y cuándo conviene usarlos, y los modos de despliegue con su trampa: el modo completo borra lo que no está en la plantilla, y eso incluye lo que creó otro equipo.

Resultados de aprendizaje

Al finalizar podrás:

1. Elegir entre Bicep y herramientas de terceros con criterio.
2. Estructurar el código por ciclo de vida y por ámbito.
3. Usar pilas de despliegue para gestionar el ciclo de vida completo.
4. Aplicar módulos verificados sin perder control.
5. Evitar los tres errores que borran recursos en producción.

Conceptos centrales

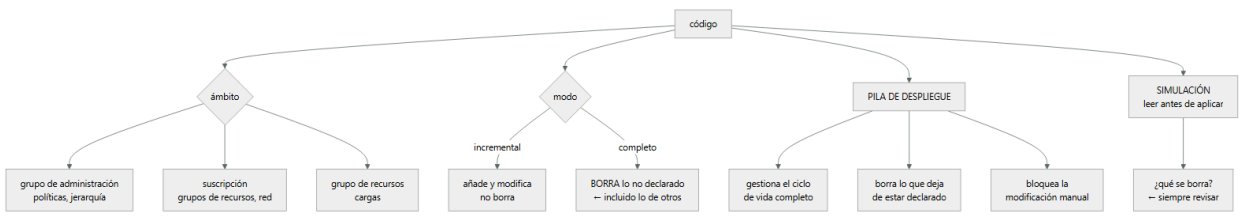
Concepto	Comprensión verificable
Bicep	Lenguaje declarativo que compila a plantillas nativas de Azure, con módulos, tipos y validación previa.
ámbito del despliegue	Nivel al que se despliega: grupo de administración, suscripción, grupo de recursos o inquilino.
modo de despliegue	Incremental añade y modifica; completo además borra lo que no está declarado.
pila de despliegue	Recurso que gestiona un conjunto como unidad, con capacidad de borrar lo que deja de estar declarado.
simulación de cambios	Previsualización de lo que el despliegue creará, modificará o borrará. Se lee antes de aplicar.
módulo verificado	Módulo mantenido por el proveedor con valores por defecto revisados y opciones de seguridad activadas.

Modelo mental

La arquitectura Azure nace del tenant y las políticas heredables, y llega al workload mediante identidades administradas y redes privadas.

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. Bicep, y cuándo no

Azure tiene su lenguaje propio y también funciona con herramientas multinube. La elección importa menos de lo que parece y más de lo que se cree.

```
BICEP
  + no hay estado que gestionar: el estado es Azure
  + soporte inmediato de recursos nuevos
  + validación y simulación de cambios nativas
  + módulos verificados mantenidos por el proveedor
  − solo Azure
  − la gestión del ciclo de vida completo exige pilas de
    despliegue; el despliegue normal no borra nada

HERRAMIENTA MULTINUBE (Terraform o similar)
  + una sola forma de trabajar para varias nubes
  + el estado permite saber qué gestiona y borrar lo que
    sobra
  + ecosistema de módulos enorme
  − hay que gestionar y proteger el estado    clase 100
  − los recursos nuevos tardan en estar soportados
  − y la portabilidad del CÓDIGO es menor de lo que se
    espera: los recursos son específicos    clase 158
```

Y el criterio honesto:

```
¿la organización usa varias nubes de verdad y el equipo ya
domina una herramienta multinube?
  → úsala también aquí; la coherencia vale más que la
    diferencia técnica

¿es una organización solo Azure, o el equipo empieza?
  → Bicep; menos piezas que operar

¿hay que declarar políticas y jerarquía?
  → cualquiera de las dos sirve, y conviene que sea la
    misma que el resto                        clase 217
```

Y una advertencia sobre la mezcla:

```
usar las dos a la vez sobre los mismos recursos produce
conflictos: cada una cree que gestiona algo que la otra
modifica
→ si conviven, que sea con fronteras claras: quién gestiona
  qué, escrito                                  clase 183
```

Los ámbitos, que hay que entender antes de estructurar:

```
INQUILINO                  grupos de administración
GRUPO DE ADMINISTRACIÓN    políticas, asignaciones
SUSCRIPCIÓN                grupos de recursos, red, roles
GRUPO DE RECURSOS          la mayoría de los recursos

→ un despliegue tiene un ámbito, y desde él puede desplegar
  módulos en ámbitos inferiores
→ estructurar el código por ámbito es lo que permite
  desplegar la jerarquía y las cargas por separado
```

2. Estructura por ciclo de vida

El error de estructura es el mismo de la clase 207: una plantilla enorme con todo.

```
✗ UNA PLANTILLA QUE LO CONTIENE ABSOLUTAMENTE todo
  cada despliegue toca todo
  un fallo bloquea el conjunto
  y los tiempos de despliegue crecen sin control

✓ SEPARADO POR CICLO DE VIDA
  jerarquía y políticas          cambian al trimestre
  red y conectividad             cambian al mes
  datos: bases, almacenes        cambian poco, NUNCA se
                                 borran con el código
  cargas: aplicaciones           cambian a diario
```

```
→ y los datos SIEMPRE en su propio despliegue, con
  bloqueos de borrado                          clase 217
```

Y las reglas que evitan los problemas frecuentes:

```
SIN VALORES INCRUSTADOS POR ENTORNO
  parámetros por entorno, en ficheros separados

NOMBRES DERIVADOS, NO FIJOS
  y con la unicidad global resuelta donde haga falta
  → varios servicios exigen nombre único en todo Azure
  → un nombre fijo funciona en el primer entorno y falla en
    el segundo

SIN SECRETOS EN PARÁMETROS
  referencia al almacén de secretos, no el valor
  → y los valores de salida NUNCA devuelven secretos: el
    historial de despliegues los conserva

IDENTIDADES Y PERMISOS DECLARADOS
  la asignación de papel es un recurso más, y debe estar
  en el código con su ámbito exacto            clase 218
```

Los módulos verificados, que resuelven un problema real:

```
QUÉ SON
  módulos mantenidos por el proveedor, con los valores por
  defecto revisados y las opciones de seguridad activadas

QUÉ RESUELVEN
  exactamente la ley 26: los valores por defecto de los
  recursos están elegidos para que la demostración
  funcione, y estos módulos los cambian por los de
  producción
  → cifrado, sin acceso público, diagnóstico, versión
    mínima de TLS, identidad administrada

CUÁNDO NO USARLOS
  cuando traen mucho más de lo que hace falta y hay que
  entender todo lo que despliegan
  cuando su versión va por detrás de lo que se necesita

→ y en cualquier caso: FIJAR LA VERSIÓN
  un módulo sin versión fija cambia bajo los pies
                                        clase 106, ley 25
```

Y una práctica que ahorra revisiones:

```
la plantilla de servicio nuevo usa los módulos verificados
y ya cumple las políticas de la organización
→ el carril fácil cumple solo                  clase 171
→ y las funciones de aptitud casi nunca fallan
```

3. Modos, pilas y lo que borra recursos

Aquí están los tres errores que borran cosas en producción.

```
MODO INCREMENTAL (el predeterminado)
  crea lo que falta, modifica lo que difiere
  NO borra nada de lo que no está en la plantilla
  → seguro, y deja recursos huérfanos acumulándose
                                                 ley 25

MODO COMPLETO
  además, BORRA todo lo que exista en el grupo de recursos
  y no esté declarado
  → resuelve los huérfanos
  → y borra lo que creó otro equipo en ese grupo
  → y lo que se creó a mano durante un incidente

  ERROR 1   usar modo completo sobre un grupo compartido
  ERROR 2   usar modo completo con una plantilla parcial
            (por ejemplo, tras un error de compilación que
             dejó fuera medio fichero)
```

Y el tercero, el más frecuente:

```
ERROR 3   no leer la simulación de cambios
  la herramienta dice exactamente qué va a crear, modificar
  y BORRAR
  → y en una canalización, ese resultado debe publicarse y
    revisarse; si incluye borrados en producción, se para
  → una puerta automática: «si hay borrados y el entorno es
    producción, exige aprobación»              clase 106
```

Las pilas de despliegue, que resuelven el ciclo de vida sin la brutalidad del modo completo:

```
QUÉ APORTAN
  gestionan un conjunto de recursos como una unidad
  saben qué recursos les pertenecen
  al desplegar, BORRAN o DESVINCULAN lo que deja de estar
    declarado
  y pueden BLOQUEAR la modificación manual de lo que
    gestionan

→ es lo más parecido al estado de una herramienta
  multinube, sin gestionar un fichero de estado

LA DECISIÓN QUE HAY QUE TOMAR AL CREARLAS
  qué hacer con lo que se quita de la plantilla
    borrar          para cargas
    desvincular     para datos ← lo prudente
  y qué bloqueo aplicar a lo gestionado
    ninguno · solo lectura · sin borrado
```

Y la comprobación que hay que hacer antes de usarlas en producción:

```
probar en un entorno inferior
  quitar un recurso de la plantilla y desplegar
  → comprobar que hace lo que se espera
  → y que los datos NO se borran                    ley 22
```

La canalización de infraestructura, con las puertas que hacen falta:

```
1  compilar y validar la plantilla
2  comprobaciones de política: ¿el despliegue cumpliría?
   → hay herramientas que evalúan la política ANTES
   → evita el rechazo en el momento de desplegar
3  SIMULACIÓN de cambios, publicada
4  puerta: si hay borrados en producción → aprobación
5  desplegar con identidad federada, sin secretos
                                              clase 218
6  comprobar que el resultado es el esperado
7  y en caso de fallo, saber cómo se vuelve atrás
```

Y una nota sobre la vuelta atrás:

```
volver atrás en infraestructura NO es volver a desplegar la
versión anterior sin más
  un recurso borrado no se recupera desplegando otra vez
  un cambio de tamaño de base no siempre es reversible
  y algunos cambios exigen recrear
→ por eso los recursos con datos llevan bloqueo y política
  de retención, y el borrado se trata como irreversible
                                              clase 166
```

4. Revisar, probar y retirar

Qué se revisa en un cambio de infraestructura, que no es lo mismo que en un cambio de código:

```
¿qué se borra? ← lo primero
¿qué se recrea? (un cambio de nombre o de ubicación
  destruye y crea)
¿qué ámbito tiene cada asignación de permisos? clase 218
¿hay acceso público en algo?
¿está el diagnóstico enrutado?
¿tiene etiquetas obligatorias?                clase 214
¿cuánto va a costar al mes?
  → algunos recursos son muy caros y se aprueban solos en
    una plantilla
```

Y la última merece automatismo:

```
estimación de coste del cambio, publicada en la revisión
→ hay herramientas que la calculan a partir de la
  simulación
→ y evita la sorpresa del mes siguiente        clase 214
```

Las pruebas de infraestructura, que son distintas de las de código:

```
VALIDACIÓN         la plantilla es correcta
SIMULACIÓN         el cambio es el esperado
DESPLIEGUE EN
  ENTORNO EFÍMERO  se despliega de cero y funciona
  → esto es lo que detecta las dependencias no declaradas
    y los recursos que alguien creó a mano y nadie metió
    en el código                              clase 104
PRUEBAS DE
  COMPORTAMIENTO   ¿el almacén rechaza el acceso público?
                   ¿la subred no llega a internet?
  → las pruebas negativas de las clases 217 a 219
```

Y el desplegar de cero es la más valiosa:

```
si el entorno completo no se puede desplegar desde el
código en un entorno vacío, el código NO describe el
sistema
→ y en un desastre, no sirve                    clase 215
→ conviene hacerlo periódicamente, no una vez
```

La retirada, que es lo que el código como declaración permite y casi nadie usa:

```
quitar el recurso de la plantilla y desplegar
  con pila de despliegue → se borra o desvincula
  con modo incremental   → NO se borra, y queda huérfano

y la comprobación periódica que encuentra los huérfanos
  inventario real frente a lo declarado
  → lo que existe y no está en ningún código es un hallazgo
                                        ley 25, clase 253
```

Y la lista de comprobación de la clase:

- la herramienta elegida es coherente con el resto de la organización
- el código está separado por ciclo de vida y por ámbito
- los datos tienen su propio despliegue, con bloqueos
- no hay valores incrustados por entorno
- los nombres son derivados y resuelven la unicidad global
- no hay secretos en parámetros ni en salidas
- las asignaciones de permisos están en el código, con su ámbito exacto
- los módulos verificados están con versión fija
- el modo completo no se usa sobre grupos compartidos
- la simulación de cambios se publica y se revisa
- hay puerta de aprobación si hay borrados en producción
- las pilas de despliegue se probaron en un entorno inferior
- el entorno se despliega de cero periódicamente
- hay estimación de coste en la revisión
- se comparan periódicamente el inventario real y lo declarado

Y el cierre que enlaza con la clase siguiente: con la base declarada, queda desplegar las cargas. Las tres opciones de cómputo gestionado de Azure —y cuándo conviene cada una— son la materia de la clase 221.

Ejemplo trabajado

CloudShop pasa su infraestructura de Azure a código. Lo que sigue es el despliegue que borró una base de datos de preproducción, la comparación entre inventario real y código declarado, y las pilas de despliegue que resolvieron los huérfanos.

El punto de partida:

```
recursos en el inventario                    14.200
  declarados en algún código                  4.100  29 %
  creados a mano                             10.100  71 %
```

```
plantillas existentes                              31
  con valores incrustados por entorno              24
  con secretos en parámetros                        7
  con nombres fijos                                19
    → y por eso solo funcionaban en un entorno
```

El incidente: el modo completo, semana 3.

```
qué pasó
  un equipo migró su carga a Bicep
  desplegó en modo COMPLETO sobre el grupo de recursos de
  preproducción
  su plantilla declaraba la aplicación y el almacén

  el grupo contenía además
    una base de datos de preproducción, creada a mano
    dos cuentas de almacenamiento de otro equipo
    un espacio de trabajo de análisis

  el modo completo BORRÓ los cuatro

efectos
  la base de datos tenía copia de seguridad: recuperada en
  2 h 40
  las cuentas de almacenamiento del otro equipo: 1 se
  recuperó del borrado reversible, la otra había superado
  la ventana → 11 GB de resultados de pruebas perdidos
  el espacio de análisis: recreado, sin las consultas
  guardadas

tiempo total de recuperación                      6 h
```

Y el análisis:

```
la simulación de cambios SE HABÍA EJECUTADO
y decía claramente «se eliminarán 4 recursos»
nadie la leyó: la canalización la imprimía en un registro
de 400 líneas y pasaba al paso siguiente        ley 15

correcciones
  1  la simulación se publica como resumen legible, con los
     borrados destacados
  2  puerta automática: si hay borrados, exige aprobación
     explícita, en cualquier entorno
  3  el modo completo queda PROHIBIDO por política; se usan
     pilas de despliegue
  4  un grupo de recursos por equipo y por carga: nada
     compartido                                clase 183
  5  bloqueo de borrado en todos los recursos con datos
```

La estructura nueva:

```
infra/
  jerarquia/        ámbito: grupo de administración
    grupos de administración, iniciativas de política
    despliegue trimestral
  conectividad/     ámbito: suscripción
    centro, radios, cortafuegos, zonas DNS privadas
    despliegue mensual
  datos/            ámbito: grupo de recursos
    bases, almacenes, espacios de análisis
    con bloqueo de borrado y política de retención
    despliegue: raro, y siempre con aprobación
  cargas/           ámbito: grupo de recursos
    aplicaciones, planes, identidades, asignaciones
    despliegue diario

parámetros por entorno, en ficheros separados
nombres derivados: {carga}-{entorno}-{region}-{sufijo único}
secretos: referencias al almacén, nunca valores
```

Los módulos verificados, y lo que cambiaron.

```
se migraron 9 tipos de recurso a módulos verificados
con versión fija
```

```
lo que traían activado que las plantillas propias no tenían
  cuenta de almacenamiento
    acceso público                      desactivado
    versión mínima de TLS               1.2
    diagnóstico                         enrutado
    identidad administrada              activada
    clave gestionada por el cliente     opcional, activada
  base de datos
    acceso público                      desactivado
    conexión por punto privado          sí
    auditoría                           activada
    retención de copias                 35 días
  plan de aplicación
    versión mínima de TLS               1.2
    registro de aplicación              activado

→ 14 valores por defecto cambiados sin escribir una línea
→ y el cumplimiento de las iniciativas de la clase 217
  pasó del 62 % al 97 % en las cargas nuevas
                                                    ley 26
```

Y la decisión sobre versiones:

```
los módulos se fijan a una versión concreta
se actualizan una vez al trimestre, revisando el registro
de cambios
→ en la primera actualización, un módulo cambió un valor
  por defecto que recreaba el recurso
→ la simulación lo detectó y se planificó, en vez de
  ocurrir de madrugada                        clase 106
```

Las pilas de despliegue:

```
se adoptaron para cargas y para conectividad

configuración
  cargas         lo que se quita de la plantilla → BORRAR
                 bloqueo de lo gestionado → sin borrado
  conectividad   lo que se quita → BORRAR
                 bloqueo → solo lectura
  datos          NO se usan pilas; despliegue incremental
                 con bloqueos y aprobación

y la prueba, en preproducción
  se quitó un recurso de la plantilla de cargas
  → borrado, correctamente
  se quitó un recurso de la plantilla de datos por error
  → NO se borró, porque datos no usa pilas         ✓

efecto sobre los huérfanos
  recursos huérfanos en cargas         1.140 → 0 en 3 meses
  → al migrar cada carga a pila, lo no declarado se
    revisaba y se borraba o se declaraba
```

La comparación entre inventario y código:

```
se montó una comprobación semanal
  inventario real frente a lo declarado en el repositorio

primera ejecución
  recursos existentes y no declarados            10.100

seis meses después
  recursos existentes y no declarados               410
    de ellos, con excepción registrada              340
      · recursos gestionados por servicios (subredes de
        integración, identidades del sistema)
    sin excepción                                    70
      → 70 hallazgos, revisados uno a uno
      → 41 se declararon, 22 se borraron, 7 resultaron
        ser de un proyecto que nadie recordaba   ley 25
```

El despliegue de cero, ejecutado dos veces:

```
primer intento (mes 4)
```

```
  se desplegó el entorno de preproducción completo en una
  suscripción vacía
  falló en 6 puntos
    · una zona DNS privada que se creaba a mano
    · una asignación de papel al grupo del equipo, que
      existía desde 2022 y no estaba en el código
    · un certificado subido manualmente
    · dos reglas del cortafuegos
    · una cuota que había que pedir con antelación
  tiempo                            3 días con correcciones

segundo intento (mes 7)
  desplegado de cero, sin intervención
  tiempo                                              94 min
  → y ese número es el que se usa en el plan de
    continuidad                                    clase 215
```

El resultado:

```
recursos declarados en código               29 %        97 %
plantillas con secretos                        7           0
plantillas con nombres fijos                  19           0
recursos borrados por error                    4           0
cumplimiento de iniciativas (cargas nuevas) 62 %        97 %
huérfanos en grupos de cargas              1.140           0
tiempo de despliegue de cero             imposible      94 min
valores por defecto corregidos por
  módulos verificados                          —          14
```

La lección que esta clase deja: el incidente que borró cuatro recursos no fue por una herramienta peligrosa: la simulación había dicho exactamente qué iba a borrar, y estaba en un registro de cuatrocientas líneas que nadie leía. Y los módulos verificados corrigieron catorce valores por defecto sin escribir una línea de código, que es la forma más barata que este programa ha encontrado de tratar la ley 26.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-18-azure-production-architecture/220-bicep-deployment-stacks-y-azure-verified-modules/lab.py
```

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

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

Evidencia esperada

El artefacto principal es un plan reproducible sin secretos ni cambios inesperados. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye azure-bicep-stack 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 despliegue borra recursos que no eran suyos	Modo completo sobre un grupo de recursos compartido	Un grupo por equipo y carga, prohíbe el modo completo y usa pilas de despliegue con la política de borrado decidida.
La simulación avisaba del borrado y nadie lo vio	El resultado se imprimía en un registro larguísimo y la canalización seguía	Publica un resumen legible con los borrados destacados y pon una puerta de aprobación cuando existan.
La plantilla funciona en un entorno y falla en el siguiente	Nombres fijos y valores incrustados por entorno	Deriva los nombres del entorno y resuelve la unicidad global; parámetros en ficheros separados.
Los recursos nuevos no cumplen las políticas de la organización	Se declaran con los valores por defecto del recurso	Usa módulos verificados con versión fija; traen los valores de producción activados.
Se acumulan recursos que nadie declaró ni recuerda	El despliegue incremental nunca borra lo que deja de estar en la plantilla	Usa pilas de despliegue y compara periódicamente el inventario real con lo declarado.
En un desastre, el entorno no se puede reconstruir desde el código	Hay piezas creadas a mano que nunca entraron en el repositorio	Despliega el entorno de cero en una suscripción vacía periódicamente y corrige lo que falte.

■ 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 el modo incremental y el completo, y qué riesgo tiene el segundo?
2. ¿Qué aportan las pilas de despliegue frente al despliegue normal?
3. ¿Por qué los módulos verificados son una respuesta directa a la ley 26?
4. ¿Qué se revisa primero en un cambio de infraestructura?
5. ¿Qué detecta desplegar el entorno de cero que ninguna otra prueba detecta?

Referencias

- Microsoft (2025). Bicep documentation. <https://learn.microsoft.com/en-us/azure/azure-resource-manager/bicep/overview>
- Microsoft (2025). Deployment stacks. <https://learn.microsoft.com/en-us/azure/azure-resource-manager/bicep/deployment-stacks>
- Microsoft (2025). ARM template deployment modes. <https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/deployment-modes>
- Microsoft (2025). Azure Verified Modules. <https://azure.github.io/Azure-Verified-Modules/>

- Microsoft (2025). ARM template what-if operation. <https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/deploy-what-if>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 18 en PDF · Recorrido de Azure en PDF · Manual integral

Anterior	Índice	Siguiente
← 219 · Hub-spoke, Virtual WAN, Private Link y DNS privado	Parte 18 · Programa	221 · App Service, Functions y Container Apps en producción →

Evaluación — 220 Bicep, deployment stacks y Azure Verified Modules

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 azure-bicep-stack 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.

221 — App Service, Functions y Container Apps en producción

Parte: 18 — Azure: arquitectura empresarial y operación en producción Nivel: avanzado · Horas estimadas: 4 Laboratorio: serverless · Estado: EXECUTABLECORE

Propósito

Elegir entre las tres opciones de cómputo gestionado de Azure y configurarlas para producción, que es distinto de configurarlas para que funcionen. La clase compara servicio de aplicaciones, funciones y aplicaciones de contenedor por lo que de verdad las diferencia —modelo de escalado, arranque en frío, integración de red y coste—, y desarrolla los ajustes que separan un despliegue estable de uno que corta peticiones y se queda sin conexiones.

Resultados de aprendizaje

Al finalizar podrás:

1. Elegir entre las tres opciones con criterios comprobables.
2. Integrar las cargas en la red privada, de entrada y de salida.
3. Configurar escalado, arranque en frío y límites sin desperdiciar.
4. Desplegar con ranuras o revisiones, sin cortar peticiones.
5. Evitar el agotamiento de puertos y de conexiones bajo carga.

Conceptos centrales

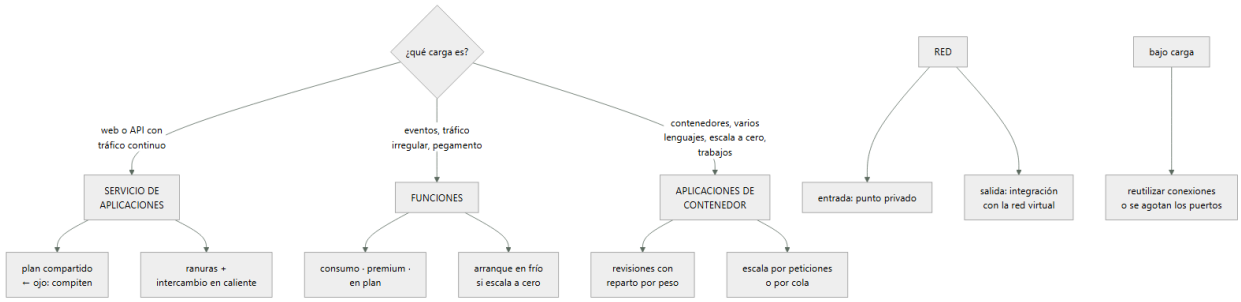
Concepto	Comprensión verificable
plan de servicio	Conjunto de recursos de cómputo compartido por varias aplicaciones. Es donde se paga y donde se compite.
ranura de despliegue	Copia de la aplicación con su propia dirección, que se intercambia con producción en caliente.
intercambio en caliente	Cambio de la ranura de preparación a producción tras calentarla, sin reiniciar.
plan de consumo	Modelo de facturación por ejecución, con escalado a cero y arranque en frío.
integración de red saliente	Mecanismo que hace que el tráfico de salida de la aplicación pase por la red virtual.
agotamiento de puertos	Falta de puertos efímeros por no reutilizar conexiones. Produce fallos intermitentes bajo carga.

Modelo mental

La arquitectura Azure nace del tenant y las políticas heredables, y llega al workload mediante identidades administradas y redes privadas.

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 opciones, y cómo elegir

Las tres ejecutan código gestionado y se diferencian en el modelo de escalado, en lo que hay que empaquetar y en el coste.

```
SERVICIO DE APLICACIONES
  aplicaciones web y APIs con proceso continuo
  + ranuras de despliegue con intercambio en caliente
  + escalado por reglas, con instancias siempre encendidas
  + integración de red madura
  – se paga el plan aunque no haya tráfico
  – varias aplicaciones en un plan COMPITEN por sus
    recursos                                  ← ver abajo

FUNCIONES
  para eventos, trabajos programados y pegamento
  + escalado por evento; en consumo, escala a cero
  + integración directa con colas, temas y almacenes
  – arranque en frío si escala a cero
  – los planes que lo evitan cuestan como una instancia

APLICACIONES DE CONTENEDOR
  contenedores sin gestionar orquestador
  + escala a cero y escalado por peticiones o por cola
  + revisiones con reparto de tráfico por peso
  + trabajos por lotes y procesos de larga duración
  – menos control que un orquestador completo
  – arranque en frío si escala a cero
```

Y el criterio, en preguntas:

```
¿es una web o API con tráfico continuo y en un lenguaje
 soportado?
  → servicio de aplicaciones

¿es código que reacciona a eventos y no está siempre
 activo?
  → funciones

¿hay contenedores, varios lenguajes, o hace falta escalar a
 cero con reparto de tráfico?
  → aplicaciones de contenedor

¿hace falta control fino del planificador, operadores o
 cargas heterogéneas?
  → Kubernetes                                clase 222
```

Y el error de elección más frecuente:

```
usar funciones para una API con tráfico continuo
  → se paga por ejecución algo que estaría más barato en un
    plan
  → y se sufre arranque en frío sin necesidad
→ el modelo de facturación por ejecución gana cuando el
  tráfico es IRREGULAR, no cuando es alto y constante
```

El plan compartido, que produce incidentes difíciles de diagnosticar:

```
varias aplicaciones en el mismo plan comparten CPU, memoria
```

```
y puertos
→ una aplicación con una fuga de memoria afecta a las
  demás
→ y el escalado del plan lo dispara cualquiera de ellas

regla   producción, en su propio plan
        y las aplicaciones de un mismo plan, del mismo
        equipo y con perfiles parecidos          clase 183
```

2. Red: entrada privada y salida integrada

Estos servicios nacen con dirección pública. Meterlos en la red privada exige dos cosas distintas que se confunden.

```
ENTRADA — punto privado
  la aplicación se alcanza por una dirección de tu red
  y el acceso público se desactiva
  → y hace falta la zona DNS privada enlazada a todas las
    redes que la llamen                       clase 219

SALIDA — integración con la red virtual
  el tráfico SALIENTE de la aplicación pasa por una subred
  → sin esto, la aplicación sale a internet aunque tenga
    punto privado de entrada
  → y por tanto no alcanza recursos privados y no pasa por
    el cortafuegos                            clase 200
```

Y la confusión que produce incidentes:

```
«ya tiene punto privado, está en la red»
  → NO: la entrada es privada y la salida sigue siendo
    pública
  → hay que configurar las dos

y el síntoma
  la aplicación no puede llegar a la base de datos privada
  o el tráfico hacia el almacén sale a internet y se paga
                                        clase 219, ley 26
```

Y los detalles de la integración de salida:

```
exige una subred DEDICADA y delegada al servicio
  con tamaño suficiente: cada instancia consume direcciones
  → dimensionar por el máximo del escalado    clase 193

y hay que decidir si CUANTO tráfico salga va por la red
  → por defecto, solo el privado; el resto sale directo
  → para forzar la salida por el cortafuegos hay que
    activarlo explícitamente
```

Y una comprobación que hay que hacer:

```
desde la aplicación, resolver el nombre de un servicio con
punto privado
  → debe devolver dirección privada
y comprobar la dirección de salida observada
  → debe ser la del cortafuegos, no una pública del
    servicio                                  ley 22
```

3. Escalado, arranque en frío y despliegue

El escalado, con la misma lección de la clase 212:

```
LA SEÑAL
  ✗ CPU, en servicios que esperan a la red
  ✓ peticiones en cola, longitud de cola, o métrica propia
  → en aplicaciones de contenedor, el escalado por cola es
    directo y es el que corresponde a un trabajador
                                              clase 210

EL RETRASO
  detección + decisión + arranque de instancia
  → minutos, igual que en la clase 212
  → y por eso el margen y el escalado programado siguen
    haciendo falta

MÍNIMO DE INSTANCIAS
```

```
  con escala a cero, la primera petición paga el arranque
  → para el camino crítico, mínimo 1 o más
  → y ahí desaparece la ventaja de coste de escalar a cero
```

El arranque en frío, con las palancas de siempre:

```
paquete pequeño y dependencias mínimas
trabajo fuera de la inicialización, salvo lo reutilizable
instancias siempre listas (se pagan)
calentamiento antes de recibir tráfico
  → y esto último es lo que hacen bien las ranuras
```

El despliegue sin cortar peticiones, que aquí tiene mecanismos propios:

```
RANURAS DE DESPLIEGUE (servicio de aplicaciones)
  se despliega en una ranura de preparación
  se CALIENTA: se piden rutas que fuercen la
    inicialización
  se intercambia con producción
  → el intercambio cambia el enrutamiento, no reinicia
  → y si algo va mal, se intercambia de vuelta en segundos

  y el detalle que decide si funciona
    la configuración con marca de «ranura» NO se
    intercambia
    → así la ranura de preparación apunta a la base de
      preparación, y producción a la suya
    → olvidarlo hace que preparación escriba en producción

REVISIONES (aplicaciones de contenedor)
  cada despliegue crea una revisión
  el tráfico se reparte por peso entre revisiones
  → despliegue escalonado nativo, con vuelta atrás
    inmediata                                clase 102

Y EN AMBOS CASOS
  parada elegante: atender la señal de terminación
  drenaje mayor que la petición más larga      clase 212
```

Y una advertencia sobre el calentamiento:

```
intercambiar sin calentar traslada el arranque en frío a
los primeros usuarios de producción
→ y el síntoma es «tras cada despliegue, unos minutos de
  latencia alta»
```

4. Lo que rompe bajo carga

El agotamiento de puertos efímeros es el fallo característico de estos servicios y se diagnostica mal.

```
QUÉ PASA
  cada conexión saliente consume un puerto efímero
  el puerto queda ocupado un tiempo tras cerrarse
  el número de puertos por instancia es limitado

  si el código abre una conexión NUEVA en cada llamada
  → bajo carga se agotan
  → y aparecen fallos intermitentes de conexión, sin
    patrón claro

LA CAUSA CASI SIEMPRE
  crear el cliente HTTP dentro de la función o del
  controlador, en vez de reutilizarlo
  → cada petición abre una conexión nueva

LA CORRECCIÓN
  cliente único y reutilizado, con agrupación de conexiones
  y para bases de datos, agrupación configurada con límite

Y EL DIAGNÓSTICO
  la métrica de conexiones salientes por instancia
  → si crece linealmente con el tráfico y no se estabiliza,
    no se están reutilizando
```

Y el otro fallo característico, ya conocido:

```
CONEXIONES A LA BASE DE DATOS
  cada instancia abre su propia agrupación
  20 instancias × 100 conexiones = 2.000
  → y la base admite 400                        clase 207
→ el límite de la agrupación se calcula por instancia,
  contando el máximo del escalado
```

Lo que hay que vigilar:

```
peticiones, errores y latencia por percentil  clase 211
instancias activas frente al máximo
conexiones salientes por instancia
tiempo de arranque y peticiones que lo pagan
uso de CPU y memoria, y REINICIOS por memoria
y en funciones: ejecuciones, retrasos de disparador y
  mensajes pendientes
```

Y una alerta específica de estos servicios:

```
«la aplicación se ha reiniciado N veces en la última hora»
  → los reinicios silenciosos por memoria o por fallo de
    la comprobación son la señal que más se pierde
                                                  ley 13
```

Y la lista de comprobación de la clase:

- la opción elegida corresponde al perfil de la carga
- producción no comparte plan con otras cargas
- la entrada es por punto privado y el acceso público está desactivado
- la salida está integrada con la red virtual
- la subred de integración está dimensionada por el máximo
- se ha comprobado que la salida pasa por el cortafuegos
- el escalado usa una señal distinta de la CPU si procede
- el mínimo de instancias cubre el camino crítico
- hay ranuras o revisiones, con calentamiento previo
- la configuración específica de ranura está marcada
- la aplicación atiende la señal de terminación
- los clientes HTTP y de base se reutilizan
- el límite de agrupación se calculó por instancia
- hay alerta de reinicios y de conexiones salientes

Y el cierre que enlaza con la clase siguiente: cuando la carga exige control fino, operadores o cargas heterogéneas, aparece Kubernetes gestionado con sus particularidades de identidad y de entrada en esta nube. Es la materia de la clase 222.

Ejemplo trabajado

CloudShop despliega su plataforma en Azure. Lo que sigue es la elección de opción por carga, el incidente de la ranura que escribió en producción, y el agotamiento de puertos que se diagnosticó como «problema de red» durante tres semanas.

La elección, carga por carga:

```
carga                perfil               elección
api de pedidos       continuo, 900 pet/s  servicio de
                                          aplicaciones
web de tienda        continuo             servicio de
                                          aplicaciones
procesador de
  eventos            por cola, irregular  aplicaciones de
                                          contenedor
trabajos programados 12/día               funciones,
                                          plan de consumo
notificaciones       picos, escala a cero funciones,
                                          plan de consumo
informes nocturnos   1/día, 40 min        trabajo de
                                          aplicaciones de
                                          contenedor
búsqueda             contenedor propio,
                     imagen específica    aplicaciones de
                                          contenedor

y lo que se descartó
```

```
  poner la api de pedidos en funciones
  → estimación: 1.840 €/mes en consumo frente a 610 € en
    un plan, con tráfico continuo
  → y arranque en frío innecesario
```

Incidente 1 · La ranura que escribió en producción.

```
síntoma   tras un despliegue de rutina, aparecieron 41
          pedidos de prueba en la base de producción
          con importes de 0,01 € y direcciones falsas

diagnóstico
  la cadena de conexión a la base estaba en la
  configuración de la aplicación
  NO estaba marcada como específica de ranura
  → al intercambiar, la configuración de preparación viajó
    a producción y la de producción a preparación
  → durante 8 minutos, antes de que alguien lo notara, las
    pruebas de humo de preparación escribieron en
    producción

corrección
  toda la configuración que apunta a recursos externos,
  marcada como específica de ranura
  y mejor: las cadenas se resuelven por referencia al
  almacén de secretos, con el nombre del secreto distinto
  por ranura
  y una comprobación previa al intercambio: la ranura de
  preparación debe apuntar a la base de preparación

y lo que se limpió
  41 pedidos borrados, con registro de la incidencia
```

Incidente 2 · «Fallos intermitentes de red», tres semanas.

```
síntoma   entre las 11:00 y las 13:00, un 2-4 % de las
          llamadas de la api de pedidos al servicio de
          precios fallaban con error de conexión
          fuera de esas horas, nunca

lo que se revisó (mal)
  reglas del grupo de seguridad          correctas
  rutas y cortafuegos                    correctos
  capturas de paquetes                   sin nada raro
  se abrió incidencia con el proveedor   3 semanas

lo que lo resolvió
  la métrica de conexiones salientes por instancia

    hora        peticiones/s   conexiones salientes
    09:00          210              240
    10:00          390              450
    11:00          720              830
    12:00        1.100            1.270   ← límite ~1.280
    14:00          380              440

  → las conexiones crecían LINEALMENTE con las peticiones
  → no se estaban reutilizando

causa
  el código creaba un cliente HTTP nuevo en cada llamada
  al servicio de precios
  cada uno abría su conexión, que quedaba ocupada tras
  cerrarse
  → a partir de cierto tráfico, no había puertos libres

corrección
  un cliente único reutilizado, con agrupación configurada
  conexiones salientes en el pico    1.270 → 46
  fallos intermitentes                 2-4 % → 0

y lo que enseñó
  la métrica que resolvió el caso estaba disponible desde
  el primer día y no estaba en ningún panel      ley 15
```

```
  → se añadió al panel estándar de servicio     clase 211
```

La red, montada en dos partes:

```
ENTRADA
  punto privado para la api y para la web
  acceso público desactivado
  zonas DNS privadas enlazadas a las 43 redes clase 219
  el tráfico público entra por la pasarela de aplicación,
  que sí es pública, con filtrado

SALIDA
  integración con la red virtual, subred dedicada /24 por
  entorno
  → dimensionada para el máximo de 30 instancias
  encaminamiento de CUANTO tráfico salga por la red
  → así pasa por el cortafuegos y se registra

y la comprobación que se añadió
  desde la aplicación, resolver el nombre de la base
  → debe dar dirección privada
  y consultar la dirección de salida observada
  → debe ser la del cortafuegos
  → en la primera ejecución, la web daba una dirección
    pública: le faltaba activar el encaminamiento completo
                                                 ley 22
```

El escalado y el despliegue:

```
api de pedidos (servicio de aplicaciones)
  escalado por peticiones en cola, no por CPU
  mínimo 3 instancias, máximo 30
  escalado programado: +8 instancias a las 10:30 en
    campañas
  ranuras: preparación y producción
    calentamiento: 6 rutas pedidas antes de intercambiar
    y tras corregir el incidente 1, comprobación previa

  peticiones con latencia alta tras despliegue
    antes del calentamiento                    ~340
    después                                       0

procesador de eventos (aplicaciones de contenedor)
  escalado por longitud de cola: 1 réplica por cada 50
    mensajes pendientes, máximo 40
  mínimo 1 réplica: la cola nunca está vacía mucho tiempo
  revisiones con reparto: 10 % durante 10 min, luego 100 %
  vuelta atrás: cambiar el peso, 15 segundos

notificaciones (funciones, consumo)
  escala a cero; arranque en frío de 400-900 ms
  aceptable: el envío es asíncrono           clase 210
  y se midió antes de aceptarlo
```

El coste, comparado:

```
api de pedidos (plan P1v3 × 3-30)             610 €    720 €
web de tienda (plan compartido con api)         —        —
  → se separó: comparten plan y compiten
  → plan propio                                          310 €
procesador de eventos                         280 €    240 €
funciones (consumo)                            90 €     71 €
trabajos de informes                           40 €     38 €
pasarela de aplicación                        320 €    320 €
puntos privados                                —       190 €  ←
inspección del cortafuegos                     —       410 €  ←
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
total                                       1.340 €  2.299 €

y las dos partidas no estimadas son las de siempre:
lo que cuesta poner las cargas en la red privada
                                            clase 216, ley 26
```

El resultado:

```
fallos intermitentes de conexión            2-4 %          0
conexiones salientes en el pico             1.270          46
pedidos de prueba en producción                41           0
peticiones lentas tras despliegue             340           0
aplicaciones compartiendo plan con
  producción                                    2           0
salida que no pasa por el cortafuegos           1           0
tiempo de vuelta atrás                    redesplegar     15 s
```

La lección que esta clase deja: tres semanas de incidencia con el proveedor y capturas de paquetes por un problema que no era de red: el código abría una conexión nueva en cada llamada y agotaba los puertos efímeros a partir de cierto tráfico. La métrica que lo resolvía estaba disponible desde el primer día. Y el incidente más embarazoso —cuarenta y un pedidos de prueba en producción— lo causó una configuración que no estaba marcada como específica de ranura, que es exactamente el tipo de detalle que funciona hasta el día del intercambio.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-18-azure-production-architecture/221-app-service-functions-y-container-apps-en-produccion/lab.py
```

El laboratorio selecciona el motor de práctica serverless 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 azure-app-platform 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 función con límites, reintentos e idempotencia. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye azure-app-platform 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
Fallos intermitentes de conexión que aparecen solo con tráfico alto	El código crea un cliente nuevo por llamada y agota los puertos efímeros	Reutiliza un cliente único con agrupación de conexiones y vigila las conexiones salientes por instancia.
La aplicación no alcanza recursos privados pese a tener punto privado	El punto privado resuelve la entrada; la salida sigue siendo pública	Configura además la integración de salida con la red virtual y comprueba la dirección de salida observada.
Tras un intercambio de ranuras, la preparación escribe en producción	La configuración con la cadena de conexión no estaba marcada como específica de ranura	Marca toda la configuración que apunte a recursos externos y comprueba antes del intercambio a qué apunta cada ranura.
Tras cada despliegue hay unos minutos de latencia alta	Se intercambia sin calentar y el arranque en frío lo pagan los usuarios	Pide rutas que fuercen la inicialización en la ranura de preparación antes de intercambiar.
Una aplicación con problemas degrada a otras sin relación	Comparten plan de servicio y por tanto recursos	Producción en su propio plan; agrupa solo aplicaciones del mismo equipo con perfiles parecidos.
Se paga mucho por una API con tráfico continuo	Se eligió el modelo de facturación por ejecución para una carga constante	El pago por ejecución gana con tráfico irregular; con tráfico alto y continuo, un plan sale más barato y evita arranques en frío.

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 decide entre las tres opciones de cómputo gestionado?
2. ¿Qué diferencia hay entre el punto privado de entrada y la integración de salida?
3. ¿Qué configuración no debe intercambiarse al usar ranuras y por qué?
4. ¿Cuál es la causa habitual del agotamiento de puertos y cómo se detecta?
5. ¿Por qué el calentamiento previo al intercambio es parte del despliegue?

Referencias

- Microsoft (2025). Azure App Service deployment slots. <https://learn.microsoft.com/en-us/azure/app-service/deploy-staging-slots>
- Microsoft (2025). Azure Functions hosting options. <https://learn.microsoft.com/en-us/azure/azure-functions/functions-scale>
- Microsoft (2025). Azure Container Apps revisions and traffic splitting. <https://learn.microsoft.com/en-us/azure/container-apps/revisions>
- Microsoft (2025). Integrate your app with an Azure virtual network. <https://learn.microsoft.com/en-us/azure/app-service/overview-vnet-integration>
- Microsoft (2025). Troubleshooting SNAT port exhaustion. <https://learn.microsoft.com/en-us/azure/load-balancer/troubleshoot-outbound-connection>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 18 en PDF · Recorrido de Azure en PDF · Manual integral

Anterior	Índice	Siguiente
← 220 · Bicep, deployment stacks y Azure Verified Modules	Parte 18 · Programa	222 · AKS, workload identity, ingress y GitOps →

Evaluación — 221 App Service, Functions y Container Apps en producción

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 azure-app-platform 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.

222 — AKS, workload identity, ingress y GitOps

Parte: 18 — Azure: arquitectura empresarial y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: kubernetes · Estado: EXECUTABLECORE

Propósito

Operar Kubernetes gestionado en Azure con las particularidades que lo diferencian de lo visto en la clase 213: la identidad de carga federada con el directorio, el modelo de red —que en esta nube tiene una decisión irreversible sobre el consumo de direcciones—, la entrada gestionada y la reconciliación desde el repositorio. Y la pregunta de siempre antes de empezar: ¿hace falta, o basta con lo de la clase 221?

Resultados de aprendizaje

Al finalizar podrás:

1. Decidir si el clúster compensa frente a las opciones gestionadas.
2. Elegir el modelo de red conociendo su consumo de direcciones.
3. Dar identidad a las cargas con federación, sin secretos.
4. Configurar la entrada gestionada con certificados y filtrado.
5. Planificar actualizaciones de versión y de complementos.

Conceptos centrales

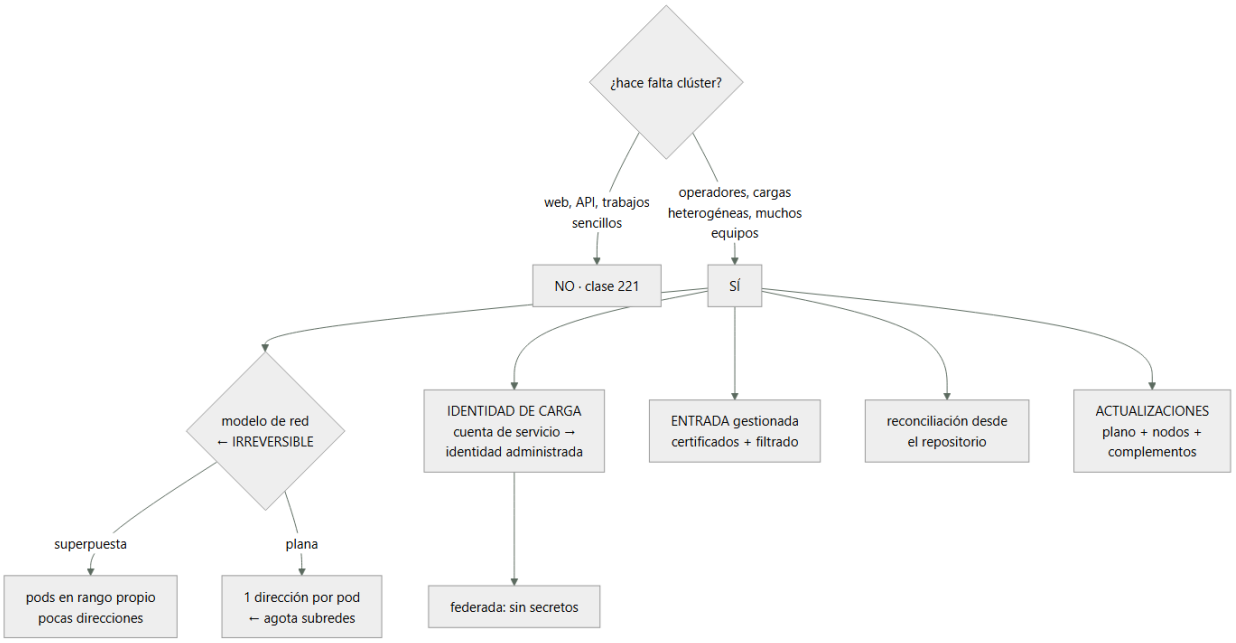
Concepto	Comprensión verificable
identidad de carga	Cuenta de servicio del clúster federada con una identidad administrada del directorio. Sin secretos.
modelo de red superpuesta	Los pods usan un rango propio no enrutado; solo los nodos consumen direcciones de la subred.
modelo de red plana	Cada pod recibe una dirección de la subred. Consume muchísimas direcciones.
controlador de entrada gestionado	Entrada operada por la plataforma, con certificados y filtrado integrados.
grupo de nodos	Conjunto de nodos con el mismo tamaño y configuración. Se actualizan y escalan por grupo.
canal de actualización	Política que decide cuándo se aplican versiones nuevas de plano de control y de nodos.

Modelo mental

La arquitectura Azure nace del tenant y las políticas heredables, y llega al workload mediante identidades administradas y redes privadas.

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. ¿Hace falta, y con qué configuración?

La pregunta de la clase 213 sigue siendo la primera, y en Azure tiene una respuesta adicional: las aplicaciones de contenedor cubren muchos casos que antes exigían clúster.

```
NO HACE FALTA SI
  son contenedores web y trabajadores por cola
  → las aplicaciones de contenedor lo hacen con escala a
    cero y reparto por revisión                clase 221
  el equipo es pequeño
  no hay cargas con requisitos especiales

SÍ COMPENSA SI
  hay operadores o recursos propios que gestionar
  hay cargas heterogéneas: GPU, procesos largos, lotes con
    prioridades
  varios equipos necesitan autonomía sobre una base común
  o ya se opera Kubernetes en otra nube y la coherencia
    vale                                       clase 158
```

Y las decisiones de configuración que se toman al crear y cuestan corregir:

```
1  MODELO DE RED           ← irreversible sin recrear
2  MODO DE IDENTIDAD       ← se puede migrar, con trabajo
3  PLANO DE CONTROL PRIVADO O PÚBLICO
4  GRUPOS DE NODOS: uno de sistema, y los de carga aparte
5  CANAL DE ACTUALIZACIÓN
```

El modelo de red, que es la decisión irreversible:

```
RED PLANA (cada pod una dirección de la subred)
  + los pods son alcanzables directamente desde la red
  + útil si algo externo tiene que llamar a un pod concreto
  − CONSUME MUCHÍSIMAS DIRECCIONES
    dimensionado = nodos máximos × pods por nodo
    40 nodos × 110 pods = 4.400 direcciones
    → una subred /22 se queda corta
  − y el número de pods por nodo se fija al crear el grupo

RED SUPERPUESTA (pods en rango propio)
  + los nodos consumen direcciones; los pods, no
    40 nodos = 40 direcciones
  + el rango de pods puede reutilizarse entre clústeres
  − los pods no son alcanzables directamente desde fuera
    → hay que salir por servicios y por la entrada
```

```
→ la superpuesta es la elección razonable salvo requisito
  concreto                                    clase 193
→ y cambiar de una a otra exige recrear el clúster
```

Y una decisión relacionada:

```
el plano de control puede ser público o privado
  privado    solo alcanzable desde la red; más seguro
  y entonces la canalización debe alcanzarlo
    → agente autoalojado en la red, o punto privado del
      plano de control
  → decidirlo antes evita rehacer la canalización
```

2. Identidad de carga, sin secretos

El mecanismo es el mismo de la clase 213, con las piezas de Azure.

```
CÓMO FUNCIONA
  el clúster expone un emisor de testigos
  se declara una CREDENCIAL FEDERADA en una identidad
    administrada, atada a
      el emisor del clúster
      el ESPACIO DE NOMBRES
      y el NOMBRE de la cuenta de servicio
  la cuenta de servicio se anota con el identificador de
    esa identidad
  el pod recibe un testigo y lo intercambia

→ y los permisos se asignan a esa identidad, en el ámbito
  del recurso                                 clase 218
```

Y los patrones que hay que dejar atrás:

```
✗ IDENTIDAD DEL NODO
  permisos en la identidad de la máquina
  → todos los pods del nodo los tienen
  → y hay que bloquear el acceso de los pods al servicio
    de metadatos, o los piden aunque tengan la suya

✗ SECRETOS DE KUBERNETES CON CREDENCIALES
  un objeto de tipo secreto está codificado, no cifrado
  → activar el cifrado del almacén del plano de control
  → y para secretos reales, obtenerlos del almacén externo
    en ejecución con un controlador, no guardarlos
```

Y el error de configuración que repite el de las clases 206 y 218:

```
la credencial federada debe atar emisor, espacio de nombres
Y cuenta de servicio
✗ atar solo el emisor
  → cualquier cuenta de servicio de cualquier espacio de
    nombres obtiene esa identidad

prueba negativa
  desde un pod de otro espacio de nombres, intentar obtener
  el testigo y usarlo
  → debe fallar                                  ley 22
```

Y la separación por equipos, con lo que aporta cada pieza:

```
espacio de nombres por equipo o carga
cuotas de recursos por espacio            clase 213
política de admisión, con mensajes que expliquen
  → qué imágenes se admiten: del registro propio y firmadas
  → sin privilegios, sin red del anfitrión
  → etiquetas obligatorias                clase 214
política de red con denegación por defecto
  → y hay que activarla al crear: por defecto, todo pod
    habla con todo pod                      ley 26
```

3. Entrada, salida y complementos

La entrada, con la elección que hay que hacer:

```
CONTROLADOR GESTIONADO POR LA PLATAFORMA
  + certificados, filtrado y escalado integrados
```

```
  + menos que operar
  − menos control de configuración fina

CONTROLADOR PROPIO EN EL CLÚSTER
  + control total, y portable a otras nubes  clase 158
  − hay que operarlo, escalarlo y actualizarlo
  − y gestionar certificados con un componente más

→ si el objetivo es reducir lo que se mantiene, el
  gestionado; si la portabilidad es un requisito real, el
  propio                                          ley 23
```

Y lo que hay que resolver en cualquiera de los dos:

```
certificados: emisión y renovación automáticas
  y ALERTA POR ANTIGÜEDAD, no solo por caducidad
                                                clase 196
filtrado de aplicación delante                clase 209
plazos y drenaje coherentes                   clase 212
y la entrada NO debe ser el único control de acceso:
  la autorización entre servicios va aparte     clase 201
```

La salida, que en un clúster es fácil de dejar abierta:

```
por defecto, los pods salen a internet por la dirección
del balanceador de salida
→ y eso es salida sin control                     ley 26

lo que hay que hacer
  ruta 0.0.0.0/0 hacia el cortafuegos           clase 219
  y política de red que restrinja a dónde puede salir cada
    espacio de nombres                          clase 200
  → y comprobar con una prueba negativa: sacar datos a un
    destino no declarado debe fallar
```

Los complementos, que son el trabajo continuo:

```
los habituales
  controlador de entrada
  proveedor de secretos desde el almacén externo
  autoescalado de nodos
  recolección de métricas y registros
  política de admisión
  controlador de certificados
  y los operadores de cada equipo

y cada uno
  tiene su matriz de compatibilidad con la versión del
    clúster
  se actualiza en su propio calendario
  y puede bloquear la actualización del clúster  ley 23

→ los que la plataforma ofrece como complemento gestionado
  se actualizan con el clúster, y eso vale mucho
→ los instalados a mano, no                     clase 213
```

Y la decisión práctica:

```
usa complementos gestionados donde existan
y para el resto, mantén la matriz de compatibilidad
escrita y comprobada antes de cada actualización
```

4. Actualizaciones y operación

Las actualizaciones tienen aquí tres piezas que se planifican por separado.

```
1  PLANO DE CONTROL
   se actualiza primero; la ventana de soporte es corta
   → 2 o 3 veces al año, obligatorio              ley 25

2  GRUPOS DE NODOS
   se actualizan después, por grupos
   con acordonado y drenaje, respetando el presupuesto de
   interrupción                                 clase 213
   → y hay que tener capacidad de sobra para el reemplazo
```

```
3  IMAGEN DE LOS NODOS
   parches del sistema operativo, más frecuentes
   → conviene automatizarlos con ventana de mantenimiento
```

Y las decisiones que evitan sorpresas:

```
CANAL DE ACTUALIZACIÓN
  manual        control total, y se olvida            ley 25
  parches       aplica parches automáticamente ← razonable
  estable       sube versión menor automáticamente
  → con VENTANA DE MANTENIMIENTO declarada, para que no
    ocurra en hora punta

Y LA COMPROBACIÓN PREVIA
  interfaces retiradas en los manifiestos
  compatibilidad de complementos
  → y un clúster de prueba GENERADO DEL MISMO CÓDIGO
    → en la clase 213, la prueba pasó porque el clúster de
      prueba no era idéntico
```

La reconciliación desde el repositorio, con la misma alerta imprescindible:

```
el estado deseado en el repositorio; un controlador lo
aplica y corrige la deriva                    clase 213

y la alerta
  «la sincronización lleva N minutos sin completarse»
  → un bucle parado no da error: deja de aplicar   ley 13
```

Y el modo automático de la plataforma, que conviene conocer:

```
hay una variante gestionada de reconciliación como
complemento
  + menos que operar
  − menos control sobre versiones y extensiones
→ misma decisión que con los demás complementos
```

Lo que hay que vigilar, además de lo de la clase 213:

```
versión del clúster y días hasta el fin de soporte ley 25
nodos con imagen desactualizada
antigüedad de la sincronización
direcciones libres en la subred de nodos
  → si se agota, no se puede escalar          clase 193
cuota de cómputo frente al máximo del escalado
  → la cuota es de la suscripción y se topa    clase 217
y reinicios por memoria y estrangulamiento por CPU
                                                clase 213
```

Y la lista de comprobación de la clase:

- ■ la decisión de usar clúster está justificada frente a las opciones gestionadas
- ■ el modelo de red se eligió contando direcciones
- ■ el plano de control es privado, y la canalización lo alcanza
- ■ hay grupo de nodos de sistema separado de los de carga
- ■ las cargas usan identidad federada, no la del nodo
- ■ los pods no pueden alcanzar el servicio de metadatos
- ■ la credencial federada ata emisor, espacio de nombres y cuenta de servicio
- ■ los secretos vienen del almacén externo en ejecución
- ■ la política de red está activada, con denegación por defecto
- ■ la salida pasa por el cortafuegos y está restringida
- ■ los certificados tienen alerta por antigüedad
- ■ se usan complementos gestionados donde existen
- ■ hay matriz de compatibilidad de los instalados a mano
- ■ el canal de actualización y la ventana están decididos
- ■ el clúster de prueba se genera del mismo código
- ■ hay alerta de antigüedad de la sincronización
- ■ se vigilan direcciones libres y cuota frente al máximo

Y el cierre que enlaza con la clase siguiente: con el cómputo resuelto, quedan los datos, que en Azure ofrecen tres familias con modelos de consistencia distintos y una de ellas con cinco niveles a elegir. Es la materia de la clase 223.

Ejemplo trabajado

CloudShop monta su clúster en Azure para las cargas del equipo de datos y el buscador. Lo que sigue es la decisión del modelo de red que evitó recrear el clúster a los cuatro meses, y los dos incidentes del primer año.

La decisión de adoptar, con el método de siempre:

```
lo que se pedía
  2 cargas con GPU para el equipo de datos
  un operador de base vectorial que ese equipo ya usa
  trabajos por lotes con prioridades
  el buscador, con imagen propia y ajustes de sistema

lo que ya estaba resuelto
  web, API y trabajadores por cola, en aplicaciones de
  contenedor                                  clase 221

decisión
  clúster SOLO para esas 4 cargas
  las 11 aplicaciones de contenedor se quedan donde están
  motivo   funcionan, y moverlas añade trabajo sin
           beneficio                            ley 23
```

El modelo de red: el cálculo que decidió.

```
el equipo iba a elegir red plana «porque es lo que usamos
en la otra nube»

el cálculo, hecho antes
  nodos máximos previstos                        34
  pods por nodo (valor por defecto)             110
  direcciones necesarias con red plana        3.740
  más el margen de despliegue escalonado      ×1,5
  ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  total                                      ~5.610
  → una subred /19

  con red superpuesta
  nodos máximos                                  34
  direcciones necesarias                         34
  más margen                                    ~60
  → una subred /26 basta

y el contexto
  el plan de direcciones asignaba /21 por radio
                                        clases 193, 219
  → con red plana, el clúster se habría comido más de la
    mitad del radio y habría bloqueado el crecimiento

decisión   red superpuesta
y lo que se comprobó antes
  ¿algo externo necesita llamar a un pod concreto?
  → no; todo pasa por servicios y por la entrada
  → sin ese requisito, la superpuesta no quita nada

y el coste de equivocarse
  cambiar el modelo exige RECREAR el clúster
  → migrar 4 cargas, sus datos y sus operadores: semanas
                                                 ley 14
```

Incidente 1 · Todos los pods podían hablar con todos, mes 2.

```
se descubrió en la revisión de seguridad, no en un incidente

la política de red NO estaba activada
  → es una opción que se activa al crear el clúster
  → y por defecto está desactivada                ley 26

qué implicaba
  un pod del espacio de nombres de pruebas del equipo de
  datos podía llamar al buscador y a la base vectorial
  y el alcance desde cualquier pod comprometido era el
  clúster entero                              clase 189
```

```
y lo peor
  activarla exige recrear el clúster en algunas
  configuraciones
  → en este caso se pudo activar sin recrear, con suerte

corrección
  política de red activada
  denegación por defecto entre espacios de nombres
  y reglas explícitas: 9 pares permitidos, de 42 posibles
  alcance desde un pod comprometido       clúster → 2
                                          servicios

y una comprobación añadida a la creación de clústeres
  función de aptitud: ningún clúster sin política de red
                                                clase 190
```

Incidente 2 · La actualización que no cabía, mes 9.

```
situación   la versión entraba en fin de soporte
            se lanzó la actualización de los grupos de nodos

qué pasó
  la actualización reemplaza nodos: acordona, drena y crea
  uno nuevo
  para crear el nuevo hacía falta cuota de cómputo
  → la suscripción estaba al 96 % de su cuota para esa
    familia de máquinas                          clase 217
  → el nodo nuevo no se pudo crear
  → la actualización quedó a medias: 3 nodos acordonados y
    drenados, sin sustituto

  capacidad efectiva                          -22 %
  y las cargas con GPU no tenían dónde ir: su grupo tenía 2
  nodos y uno estaba acordonado
  duración                                    4 h 20

corrección inmediata
  se descordonó lo drenado y se pidió ampliación de cuota
  → tardó 6 h en concederse

corrección de fondo
  cuota pedida con margen del 30 % sobre el máximo del
  escalado
  alerta al 80 % de la cuota                    clase 262
  y regla: no se actualiza si el margen de cuota es menor
  que un nodo del grupo más grande
```

La identidad, montada bien desde el principio:

```
7 identidades administradas, una por carga
credencial federada atando emisor + espacio de nombres +
  cuenta de servicio
permisos asignados en el ámbito del RECURSO   clase 218
acceso de los pods al servicio de metadatos: bloqueado
secretos desde el almacén externo, montados en ejecución

pruebas negativas
  ✓  pod de «datos-pruebas» intentando obtener la identidad
     de «buscador»                             denegado
  ✓  pod leyendo credenciales del nodo          bloqueado
  ✓  cuenta de servicio sin anotación             sin
                                                identidad
  ✗  un pod del espacio «datos» podía leer un almacén de
     producción
     → la asignación se había hecho en el ámbito del grupo
       de recursos, no del almacén
     → corregido                                clase 218
```

La entrada y los complementos:

```
entrada     controlador gestionado por la plataforma
            motivo: la portabilidad no era requisito y hay
            menos que mantener                    ley 23
            certificados automáticos, con alerta por
            antigüedad                          clase 196
```

```
            filtrado delante                    clase 209

complementos
  gestionados      métricas, registros, proveedor de
                   secretos, política, autoescalado
  a mano           el operador de base vectorial del equipo
                   de datos
                   → 1 solo, con su matriz de compatibilidad

→ frente a los 12 complementos a mano de la clase 213,
  aquí quedó 1
→ y la actualización siguiente tardó 2 h en vez de 4
```

Las actualizaciones, planificadas:

```
canal                parches automáticos
ventana              domingos 02:00-06:00
versión menor        manual, 3 veces al año, planificada

antes de cada una
  revisión de interfaces retiradas en la canalización
  clúster de prueba generado del mismo código   clase 213
  comprobación de cuota disponible
  matriz de compatibilidad del operador

segunda actualización (mes 13)
  duración                                      2 h
  incidencias                                     0
  manifiestos corregidos antes                    6
```

El resultado, al año:

```
direcciones consumidas por el clúster      ~5.610          60
  (si se hubiera elegido red plana)
alcance desde un pod comprometido      todo el clúster  2 serv.
complementos instalados a mano                  —           1
duración de una actualización              4 h 20        2 h
incidentes por cuota                            1           0
cargas movidas al clúster sin necesidad         0           0
capacidad de equipo consumida                 n/d    0,3 pers.
```

La lección que esta clase deja: la decisión que más valor tuvo se tomó antes de crear el clúster: elegir el modelo de red superpuesta ahorró más de cinco mil quinientas direcciones y evitó que el clúster se comiera medio radio del plan. Cambiarla después habría exigido recrearlo. Y el hallazgo de seguridad más grave —todos los pods podían hablar con todos— no lo produjo ningún error: era el valor por defecto, y se descubrió en una revisión, no en un incidente.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-18-azure-production-architecture/222-aks-workload-identity-ingress-y-gitops/lab.py
```

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

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

Evidencia esperada

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

Reto verificable

Construye azure-aks-platform 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 clúster consume miles de direcciones y bloquea el crecimiento de la red	Se eligió el modelo de red plana, en el que cada pod toma una dirección de la subred	Calcula nodos por pods antes de crear el clúster y elige el modelo superpuesto salvo requisito concreto; cambiarlo exige recrear.
Cualquier pod puede hablar con cualquier otro	La política de red no está activada; por defecto no lo está	Actívala al crear el clúster, con denegación por defecto y reglas explícitas, y compruébalo con una función de aptitud.
Un pod usa permisos que no le corresponden	Los permisos están en la identidad del nodo o la credencial federada solo ata al emisor	Identidad por carga, credencial atada a espacio de nombres y cuenta de servicio, y bloqueo del acceso al servicio de metadatos.
Una actualización deja nodos drenados sin sustituto	No hay cuota de cómputo para crear los nodos nuevos	Pide cuota con margen sobre el máximo del escalado, alerta al 80 % y no actualices si el margen es menor que un nodo.
La actualización del clúster se bloquea por los complementos	Muchos complementos instalados a mano, cada uno con su compatibilidad	Usa complementos gestionados donde existan y mantén escrita la matriz de los que queden.
Los cambios del repositorio dejan de aplicarse sin aviso	El bucle de reconciliación falló en silencio	Alerta por antigüedad de la sincronización, con destino a un canal con guardia.

■ 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 el modelo de red plana y el superpuesto, y por qué es irreversible?
2. ¿A qué debe atarse una credencial federada para una carga del clúster?
3. ¿Qué está permitido por defecto entre pods y cómo se cierra?
4. ¿Qué puede impedir que una actualización de nodos termine?
5. ¿Qué ventaja tienen los complementos gestionados frente a los instalados a mano?

Referencias

- Microsoft (2025). Azure Kubernetes Service: workload identity. <https://learn.microsoft.com/en-us/azure/aks/workload-identity-overview>
- Microsoft (2025). AKS networking: Azure CNI overlay. <https://learn.microsoft.com/en-us/azure/aks/azure-cni-overlay>
- Microsoft (2025). Application routing add-on and managed ingress. <https://learn.microsoft.com/en-us/azure/aks/app-routing>
- Microsoft (2025). AKS cluster upgrades and maintenance windows. <https://learn.microsoft.com/en-us/azure/aks/upgrade-cluster>
- Microsoft (2025). AKS baseline architecture. <https://learn.microsoft.com/en-us/azure/architecture/reference-architectures/containers/aks/baseline-aks>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 18 en PDF · Recorrido de Azure en PDF · Manual integral

Anterior	Índice	Siguiente
← 221 · App Service, Functions y Container Apps en producción	Parte 18 · Programa	223 · Azure SQL, Cosmos DB y consistencia distribuida →

Evaluación — 222 AKS, workload identity, ingress y GitOps

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 azure-aks-platform 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.

223 — Azure SQL, Cosmos DB y consistencia distribuida

Parte: 18 — Azure: arquitectura empresarial y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: data · Estado: EXECUTABLECORE

Propósito

Elegir y configurar el almacén de datos en Azure, donde hay una decisión que no existe en otras nubes: cinco niveles de consistencia a elegir, con su coste y su latencia. La clase compara la base relacional gestionada, la distribuida multimodelo y el resto de la familia, explica qué significa cada nivel de consistencia en la práctica, y desarrolla el modelo de capacidad —unidades de petición— que es donde se rompen los presupuestos y las latencias.

Resultados de aprendizaje

Al finalizar podrás:

1. Elegir entre base relacional gestionada y distribuida con criterios.
2. Decidir el nivel de consistencia por operación y conocer su coste.
3. Dimensionar capacidad con unidades de petición y evitar el estrangulamiento.
4. Distribuir datos entre regiones sabiendo qué se gana y qué se paga.
5. Diagnosticar los fallos característicos de cada familia.

Conceptos centrales

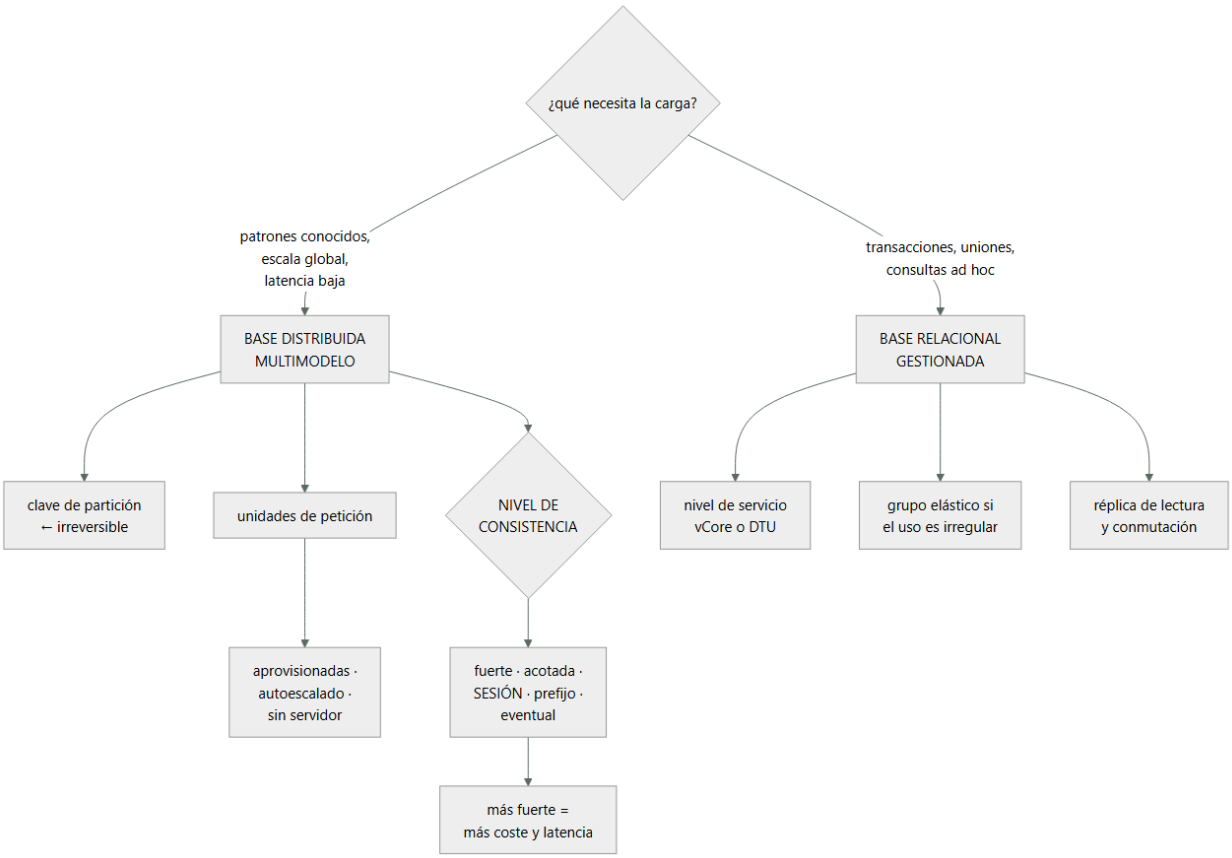
Concepto	Comprensión verificable
unidad de petición	Medida normalizada del coste de una operación. Determina capacidad, latencia y factura.
clave de partición lógica	Campo que agrupa elementos. Decide el reparto y no se puede cambiar sin migrar.
consistencia acotada	Nivel que garantiza un desfase máximo, en versiones o en tiempo.
consistencia de sesión	Nivel por defecto: un cliente ve sus propias escrituras y no retrocede.
grupo de escalado elástico	Conjunto de bases que comparten recursos, para cargas de uso irregular.
escritura multirregión	Opción que permite escribir en varias regiones. Duplica el coste y obliga a resolver conflictos.

Modelo mental

La arquitectura Azure nace del tenant y las políticas heredables, y llega al workload mediante identidades administradas y redes privadas.

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. Elegir la familia

Azure ofrece varias familias y la elección se hace igual que siempre: por los patrones de acceso.

```
BASE RELACIONAL GESTIONADA
  transacciones, uniones, consultas que no se conocen de
  antemano, informes
  + SQL, herramientas maduras, migración desde lo heredado
  + réplicas de lectura y conmutación gestionada
  – escala vertical con límite; la horizontal exige
    particionar a mano
  – y las conexiones son un recurso limitado  clase 207

BASE DISTRIBUIDA MULTIMODELO
  patrones conocidos, escala grande, latencia baja y
  distribución global
  + escala horizontal transparente
  + distribución multirregión con una casilla
  + cinco niveles de consistencia a elegir
  – el modelo se deduce de los patrones      clase 208
  – y el coste se mide en unidades de petición, que hay que
    entender

Y EL RESTO DE LA FAMILIA
  caché en memoria       para lecturas calientes
  almacén de tablas      barato y limitado
  almacenamiento de
    objetos con capas    para lo frío
  y motores gestionados de código abierto para migraciones
```

Y el criterio, que ya conocemos:

```
¿las consultas se conocen de antemano?
  no  → relacional; o distribuida MÁS un almacén analítico
  sí  → distribuida, si el volumen o la latencia lo piden

¿hace falta transacción entre entidades no relacionadas?
```

```
  → relacional

¿hace falta latencia baja en varias regiones?
  → distribuida
```

Y el patrón que resuelve la mayoría de los casos reales:

```
distribuida para el camino operativo
+ enlace analítico o flujo de cambios hacia un almacén de
  análisis para informes y exploración
→ y así ninguna de las dos hace lo que no sabe hacer
                                        clases 208, 150
```

Y una advertencia sobre la migración desde lo heredado:

```
la instancia gestionada permite migrar bases antiguas con
casi cero cambios
  + resuelve la migración
  − y traslada el modelo antiguo, con sus problemas
→ es una decisión de plazo, no de arquitectura; conviene
  escribir que es temporal, con fecha            ley 25
```

2. Los cinco niveles de consistencia

Esta es la particularidad que no existe en otras nubes: el nivel de consistencia se elige, por cuenta y por operación.

```
FUERTE
  toda lectura ve la última escritura confirmada
  coste   el más alto; y NO se puede usar con escritura
          multirregión
  latencia añadida entre regiones

OBSOLESCENCIA ACOTADA
  el desfase máximo está garantizado: N versiones o T
  segundos
  → útil cuando se puede tolerar un retraso conocido
  → y es la única que da una COTA, que es lo que suele
    hacer falta

SESIÓN  ← el valor por defecto
  el cliente ve sus propias escrituras y no retrocede
  coste   la mitad de lectura que la fuerte
  → y es lo que la mayoría de los sistemas necesitan de
    verdad                                  clase 187

PREFIJO COHERENTE
  las escrituras se leen en orden, sin huecos
  → sin garantía de recencia

EVENTUAL
  converge, sin plazo ni orden
  coste   el más bajo
```

Y las tres cosas que hay que saber para decidir:

```
1  EL COSTE DE LECTURA CAMBIA
   fuerte y acotada cuestan el DOBLE de unidades de
   petición que sesión, prefijo o eventual
   → cambiar el nivel por defecto de fuerte a sesión puede
     reducir la factura de lectura a la mitad

2  SE PUEDE RELAJAR POR PETICIÓN
   la cuenta fija el nivel por defecto
   y cada petición puede pedir uno MÁS DÉBIL, nunca más
   fuerte
   → conviene poner la cuenta en el nivel más fuerte que
     necesite alguna operación, y relajar el resto

3  LA ESCRITURA MULTIRREGIÓN LIMITA LAS OPCIONES
   con varias regiones de escritura, fuerte no está
   disponible
   → y hay que resolver conflictos, con las trampas de los
     relojes                                clase 187
```

Y la decisión práctica, con el método de la clase 187:

```
por cada operación
  reservar plaza / cobrar   → fuerte, o escritura
                              condicionada
  ver lo propio             → sesión
  listar catálogo           → eventual
  panel                     → eventual, marcado

y el nivel de la cuenta = el más fuerte que necesite alguna
→ el resto relaja explícitamente
```

Y una advertencia sobre la consistencia fuerte:

```
con regiones distribuidas, la lectura fuerte obliga a
coordinar
→ la latencia crece con la distancia
→ y por eso «pongo todo en fuerte por si acaso» es la
  decisión más cara que se puede tomar aquí       ley 26
```

3. Unidades de petición y estrangulamiento

Todo el coste, la capacidad y la latencia de la base distribuida se expresan en unidades de petición, y no entenderlas es la causa de los sustos.

```
QUÉ CONSUME
  leer un elemento por su clave: poco, y proporcional al
    tamaño
  escribir: bastante más que leer
  índices: cada índice encarece la escritura
  consulta que recorre varias particiones: MUCHO
  → y esa es la que arruina el presupuesto

LOS MODELOS DE CAPACIDAD
  APROVISIONADA        se reservan unidades por segundo
    + barata con tráfico alto y estable
    – hay que dimensionar y vigilar el estrangulamiento
  AUTOESCALADO         entre un mínimo y diez veces ese
                       mínimo
    + absorbe picos
    – precio por unidad mayor
  SIN SERVIDOR         se paga por consumo
    + ideal para desarrollo y cargas pequeñas
    – límites de caudal y de almacenamiento
```

Y el reparto por partición, que es donde se falla:

```
LAS UNIDADES SE REPARTEN ENTRE PARTICIONES FÍSICAS
  10.000 unidades y 10 particiones → 1.000 por partición

→ si el tráfico va a una sola clave, se dispone de 1.000,
  no de 10.000
→ y aparece el estrangulamiento con la base «al 10 % de
  uso»
→ es exactamente el problema de la clase 208, con otro
  nombre
```

Y el diagnóstico y las correcciones:

```
SÍNTOMA   error de petición limitada, con reintento sugerido
          latencia alta en horas concretas
          y consumo total muy por debajo de lo
          aprovisionado

QUÉ MIRAR
  unidades consumidas POR PARTICIÓN, no en total
  y las claves de partición más consumidoras

CORRECCIONES
  clave que reparta mejor            ← lo correcto
  clave sintética con sufijo         ← si ya es tarde
  caché delante para lecturas calientes
  y revisar las consultas que cruzan particiones
```

Y dos ajustes que reducen mucho el consumo:

```
POLÍTICA DE ÍNDICES
  por defecto se indexa CUALQUIER campo
```

```
  → cada escritura paga por indexar campos que nadie
    consulta                                        ley 26
  → excluir lo que no se filtra reduce el consumo de
    escritura de forma notable

TAMAÑO DEL ELEMENTO
  el coste de lectura crece con el tamaño
  → guardar campos grandes fuera y referenciarlos
```

Y la vigilancia imprescindible:

```
peticiones limitadas: alerta si superan un umbral bajo
unidades consumidas frente a las aprovisionadas
las 10 claves de partición que más consumen
y el coste por operación de negocio           clase 214
```

4. Distribución, continuidad y la familia relacional

La distribución multirregión de la base distribuida se activa fácilmente y tiene consecuencias.

```
AÑADIR UNA REGIÓN DE LECTURA
  duplica el almacenamiento y las unidades de petición
  → el coste se multiplica por el número de regiones
  + latencia baja en esa región
  + y conmutación gestionada

AÑADIR ESCRITURA MULTIRREGIÓN
  + escritura local en cada región
  − conflictos: hay que definir la política de resolución
  − y con relojes derivando, «el último gana» tiene las
    trampas conocidas                          clase 187
  → solo tiene sentido si cada dato lo escribe una sola
    región                                       ley 21

CONMUTACIÓN
  automática o manual
  → automática es cómoda y quita el control del momento
  → y volver exige que la región original se ponga al día
                                                 clase 215
```

Y el consejo:

```
empieza con una región de escritura y las de lectura que
hagan falta
y mide antes de activar escritura multirregión: casi nunca
es lo que se necesita
```

La familia relacional, con sus decisiones propias:

```
NIVEL DE SERVICIO
  el modelo por núcleos permite elegir cómputo y
  almacenamiento por separado
  → más predecible que el modelo por unidades combinadas

GRUPOS ELÁSTICOS
  varias bases comparten recursos
  → ideal para multi-inquilino con uso irregular
  → y evita pagar capacidad por cada base

RÉPLICAS DE LECTURA
  descargan informes y consultas pesadas
  → con el retraso de réplica vigilado          clase 161

CONTINUIDAD
  grupo de conmutación con nombre de escritura y de lectura
  → la aplicación usa el nombre, no la instancia
  → y el plazo real se MIDE, incluida la caché del cliente
                                          clases 195, 215

CONEXIONES
  el límite depende del nivel
  → y con cómputo que escala, hay que calcularlo por
    instancia                                    clase 221
  → o usar un intermediario de conexiones
```

Y una decisión de coste que se olvida:

```
el nivel sin servidor de la base relacional pausa la base
tras un tiempo sin uso
  + muy barato en desarrollo
  – la primera conexión tras la pausa tarda
  → excelente para entornos no productivos    clase 214
  → y una mala idea en producción con tráfico esporádico
```

Y la lista de comprobación de la clase:

- la familia elegida corresponde a los patrones de acceso
- los patrones están escritos antes del modelo
- la clave de partición reparte y está en las consultas
- el nivel de consistencia de la cuenta es el más fuerte necesario, y el resto se relaja por petición
- se conoce el coste doble de lectura de los niveles fuertes
- el modelo de capacidad corresponde al perfil de tráfico
- se vigilan las unidades por partición, no solo el total
- hay alerta de peticiones limitadas
- la política de índices excluye lo que no se consulta
- los campos grandes no viven dentro del elemento
- las regiones añadidas están justificadas por su coste
- la escritura multirregión, si existe, tiene política de conflictos escrita
- la aplicación usa nombres de conmutación, no instancias
- el límite de conexiones se calculó por instancia
- el plazo de conmutación se ha medido, no calculado

Y el cierre que enlaza con la clase siguiente: con datos y cómputo resueltos, el trabajo que no cabe en la petición se envía a procesar, y en Azure hay tres servicios de mensajería con propósitos distintos que se confunden constantemente. Es la materia de la clase 224.

Ejemplo trabajado

CloudShop elige y configura sus almacenes en Azure. Lo que sigue es la factura de la base distribuida que triplicaba lo previsto, las dos causas que la explicaban, y la decisión de consistencia que redujo el coste de lectura a la mitad.

El punto de partida, tras dos meses:

```
estimación inicial                          1.800 €/mes
factura real                                5.740 €/mes

desglose
  base distribuida (pedidos y clientes)     4.100 €
  base relacional (facturación heredada)      980 €
  caché en memoria                            340 €
  almacenamiento de objetos                   320 €
```

Causa 1 · La consistencia fuerte por defecto.

```
la cuenta se había creado con consistencia FUERTE
motivo del equipo   «es la más segura»

lo que costaba
  las lecturas fuertes consumen el doble de unidades que
  las de sesión
  lecturas al mes                             980 M
  → y el 96 % de ellas no necesitaban recencia

el análisis por operación                    clase 187
  operación                  necesita          nivel
  reservar plaza             atomicidad     escritura
                                            condicionada
  ver mi pedido              verme a mí     sesión
  listar catálogo            nada           eventual
  ver precio                 ≤ 15 min       eventual
  panel de ventas            ≤ 60 s         eventual
  cierre contable            recencia       acotada 5 s

  → ninguna operación necesitaba FUERTE
  → la más exigente era el cierre contable, con acotada
```

```
cambio
  nivel de la cuenta          fuerte → obsolescencia
                              acotada (5 s)
  y las lecturas normales piden explícitamente sesión o
  eventual

efecto
  unidades de lectura consumidas          -47 %
  coste                          4.100 € → 2.380 €
  latencia p99 de lectura           38 ms → 9 ms
```

Y la observación:

```
el nivel de consistencia se había elegido en el asistente
de creación, en 20 segundos, sin escribir ningún patrón
→ y costaba 1.720 €/mes                    ley 14, ley 26
```

Causa 2 · La política de índices por defecto.

```
por defecto se indexan TODOS los campos

el documento de pedido tenía 61 campos
  campos usados en filtros u ordenaciones           7
  campos indexados                                 61

consumo de escritura por pedido        42 unidades
tras excluir los 54 campos no consultados
                                       17 unidades

efecto
  escrituras al mes                         41 M
  coste de escritura              1.720 € → 700 €
```

Y un efecto secundario que no se buscaba:

```
dos campos indexados contenían texto largo de descripción
→ excluirlos redujo también el TAMAÑO del índice
→ almacenamiento                          410 GB → 240 GB
```

El estrangulamiento con la base al 11 %.

```
síntoma   entre las 11:00 y las 12:30, errores de petición
          limitada
          consumo total                    2.100 de 20.000
          unidades aprovisionadas

diagnóstico
  unidades por partición, no en total
  el panel de claves más consumidoras

    clave de partición          unidades consumidas
    cliente#EMPRESA-A                 1.740   ← el 83 %
    resto (41.000 claves)               360

  el cliente EMPRESA-A hacía pedidos automáticos desde su
  sistema, con ráfagas a mediodía
  su partición disponía de ~1.800 unidades de las 20.000
  → estrangulada, mientras el resto de la base estaba
    ociosa

corrección
  no se cambió la clave: habría exigido migrar
  → el sistema de EMPRESA-A pasó a agrupar sus pedidos en
    lotes con límite de ritmo
  → y se añadió una alerta: «una sola clave consume más del
    30 % del total»

y la decisión registrada
  «si aparece un segundo cliente con este perfil, se
   introduce clave sintética con sufijo y se migra»
                                       clases 208, 190
```

La distribución multirregión, evaluada y recortada.

```
se había activado escritura multirregión en 3 regiones
motivo   «para estar preparados»
```

```
lo que costaba
  almacenamiento × 3
  unidades de petición × 3
  → 2.380 € pasaban a ser ~4.900 € solo por eso

lo que aportaba
  el 98,4 % del tráfico venía de Europa occidental
  las otras dos regiones servían 1,6 %
  y la escritura multirregión obligaba a una política de
  resolución de conflictos que nadie había definido

decisión
  1 región de escritura (Europa occidental)
  1 región de lectura (Europa norte), para continuidad
                                                clase 215
  la tercera, retirada
  escritura multirregión, desactivada

coste                              4.900 € → 2.380 €
  (con la región de lectura ya incluida)
```

La base relacional, con lo suyo:

```
facturación heredada, migrada a instancia gestionada
  → decisión registrada como TEMPORAL, con revisión a 18
    meses                                        ley 25

ajustes
  nivel   por núcleos, cómputo y almacenamiento separados
  réplica de lectura para los informes
    → descargó el 61 % de las consultas del primario
  grupo de conmutación con nombres de escritura y lectura
    → la aplicación usa el nombre, no la instancia

y las conexiones
  cómputo con escalado a 30 instancias           clase 221
  agrupación por instancia                        20
  30 × 20 = 600  > 400 del nivel contratado
  → se bajó la agrupación a 12 y se añadió intermediario
  → conexiones máximas observadas                 214

y los entornos no productivos
  nivel sin servidor con pausa automática
  coste de desarrollo y preproducción     480 € → 90 €
                                                clase 214
```

El resultado:

```
coste mensual de datos                    5.740 €    2.910 €
  base distribuida                        4.100 €    1.680 €
  base relacional                           980 €      810 €
  no productivos                            480 €       90 €
unidades por escritura de pedido              42          17
latencia p99 de lectura                    38 ms        9 ms
errores de petición limitada           140/día           0
almacenamiento de la base distribuida     410 GB      240 GB
regiones activas                              3           2
escritura multirregión                       sí          no
conexiones máximas a la relacional           600         214
```

La lección que esta clase deja: de los cinco mil setecientos euros mensuales, más de la mitad venía de dos decisiones tomadas en el asistente de creación: el nivel de consistencia fuerte y la política de índices que lo indexa todo. Ninguna operación necesitaba consistencia fuerte, y siete de sesenta y un campos se consultaban. Y el estrangulamiento con la base al once por ciento de uso resultó ser un solo cliente concentrando el ochenta y tres por ciento del consumo en una partición, que es el mismo problema de la clase 208 con otro nombre.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-18-azure-production-architecture/223-azure-sql-cosmos-db-y-consistencia-distribuida/lab.py
```

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

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

Evidencia esperada

El artefacto principal es un modelo de datos ligado a patrones de acceso. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

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

■ Criterio de aceptación

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

■■ Errores frecuentes

Síntoma	Causa probable	Corrección
El coste de lectura es el doble de lo esperado	La cuenta usa un nivel de consistencia fuerte que ninguna operación necesita	Decide el nivel por operación, fija en la cuenta el más fuerte necesario y relaja el resto por petición.
Cada escritura consume muchísimas unidades	La política de índices indexa todos los campos por defecto	Excluye del índice los campos que no se filtran ni se ordenan, y saca los textos largos del elemento.
Hay estrangulamiento con la base al diez por ciento de uso	Las unidades se reparten entre particiones y el tráfico va a una sola clave	Mira el consumo por partición y las claves más consumidoras; corrige la clave, agrupa en lotes o añade sufijo sintético.
La factura se multiplica al añadir regiones	Cada región replica almacenamiento y unidades de petición	Justifica cada región con el tráfico que sirve y activa la escritura multirregión solo si cada dato lo escribe una sola región.
La base relacional agota conexiones al escalar el cómputo	Cada instancia abre su propia agrupación y el total supera el límite del nivel	Calcula la agrupación por instancia contando el máximo del escalado, o usa un intermediario de conexiones.
La aplicación no sigue la conmutación de la base	Se conecta a la instancia y no al nombre del grupo de conmutación	Usa siempre el nombre de escritura o de lectura, y mide el plazo real de conmutación incluyendo la caché del cliente.

■ Seguridad, ética y costo

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

■ Preguntas de comprobación

1. ¿Cuáles son los cinco niveles de consistencia y cuál es el valor por defecto?
2. ¿Qué niveles cuestan el doble de unidades al leer?
3. ¿Por qué puede haber estrangulamiento con la base muy por debajo de su capacidad total?
4. ¿Qué efecto tiene la política de índices por defecto sobre el coste de escritura?
5. ¿Qué obliga a resolver la escritura multirregión y cuándo tiene sentido?

Referencias

- Microsoft (2025). Azure Cosmos DB consistency levels. <https://learn.microsoft.com/en-us/azure/cosmos-db/consistency-levels>
- Microsoft (2025). Request units in Azure Cosmos DB. <https://learn.microsoft.com/en-us/azure/cosmos-db/request-units>
- Microsoft (2025). Partitioning and horizontal scaling. <https://learn.microsoft.com/en-us/azure/cosmos-db/partitioning-overview>
- Microsoft (2025). Indexing policies in Azure Cosmos DB. <https://learn.microsoft.com/en-us/azure/cosmos-db/index-policy>
- Microsoft (2025). Azure SQL Database: failover groups and elastic pools. <https://learn.microsoft.com/en-us/azure/azure-sql/database/failover-group-sql-db>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 18 en PDF · Recorrido de Azure en PDF · Manual integral

Anterior	Índice	Siguiente
← 222 · AKS, workload identity, ingress y GitOps	Parte 18 · Programa	224 · Service Bus, Event Grid y Event Hubs →

Evaluación — 223 Azure SQL, Cosmos DB y consistencia distribuida

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 azure-data-platform 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.

224 — Service Bus, Event Grid y Event Hubs

Parte: 18 — Azure: arquitectura empresarial y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: messaging · Estado: EXECUTABLECORE

Propósito

Elegir entre los tres servicios de mensajería de Azure, que se confunden constantemente porque los tres «llevan mensajes» y resuelven cosas distintas. La clase separa el bus de mensajes empresarial, el enrutador de eventos y el flujo de gran volumen por lo que cada uno garantiza, desarrolla los mecanismos propios del primero —sesiones, transacciones, detección de duplicados— y aplica la disciplina de siempre: idempotencia, cola de fallidos vigilada y reproceso probado.

Resultados de aprendizaje

Al finalizar podrás:

1. Elegir entre bus, enrutador y flujo según lo que se garantiza.
2. Usar sesiones, transacciones y detección de duplicados donde aportan.
3. Configurar bloqueo, reintentos y cola de fallidos coherentemente.
4. Consumir un flujo con posiciones y reparto por partición.
5. Operar los mensajes fallidos con alerta, diagnóstico y reproceso.

Conceptos centrales

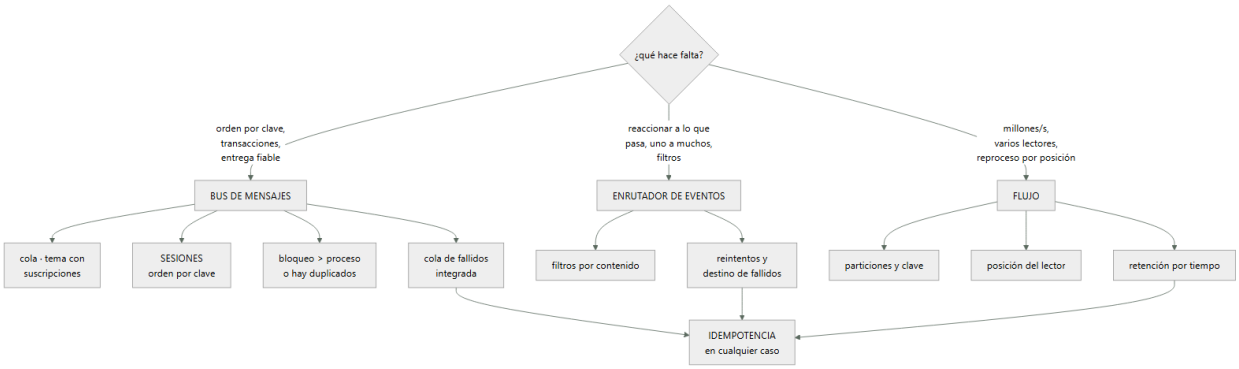
Concepto	Comprensión verificable
bus de mensajes	Colas y temas con garantías empresariales: orden por sesión, transacciones, detección de duplicados y cola de fallidos.
enrutador de eventos	Servicio que entrega eventos a destinos según filtros. Uno a muchos, con reintentos y destino de fallidos.
flujo de eventos	Registro particionado de gran volumen, con lectores independientes y reproceso por posición.
sesión	Agrupación de mensajes por clave que garantiza orden y entrega al mismo consumidor.
bloqueo del mensaje	Tiempo durante el que el mensaje tomado no se entrega a otro. Debe superar el proceso o hay duplicados.
detección de duplicados	Descarte de mensajes con el mismo identificador dentro de una ventana. No sustituye a la idempotencia.

Modelo mental

La arquitectura Azure nace del tenant y las políticas heredables, y llega al workload mediante identidades administradas y redes privadas.

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 servicios, tres propósitos

Los tres transportan mensajes y garantizan cosas distintas. Elegir mal produce sistemas que funcionan hasta que hay carga o hay que reprocesar.

```
BUS DE MENSAJES (Service Bus)
  colas y temas con suscripciones
  garantiza
    entrega al menos una vez, con bloqueo y confirmación
    ORDEN por sesión
    transacciones entre entidades del propio bus
    detección de duplicados en una ventana
    cola de fallidos integrada
    y mensajes programados y diferidos
  para   pedidos, pagos, integraciones donde perder o
         desordenar cuesta dinero
  ojo    caudal moderado; no es para millones por segundo

ENRUTADOR DE EVENTOS (Event Grid)
  entrega eventos a destinos según filtros de contenido
  garantiza
    entrega al menos una vez, con reintentos y retroceso
    destino de fallidos
    y filtros declarativos
  para   reaccionar a lo que ocurre; uno a muchos
         incluidos los eventos de la propia plataforma
  ojo    NO garantiza orden

FLUJO (Event Hubs)
  registro particionado, con lectores independientes
  garantiza
    orden DENTRO de la partición
    retención por tiempo y reproceso por posición
    varios grupos de consumidores sin duplicar el
      almacenamiento
  para   telemetría, trazas, clics, sensores: gran volumen
  ojo    el consumidor gestiona su posición; si la pierde,
         reprocesa o se salta                  clase 116
```

Y la combinación habitual, que es la que hay que reconocer:

```
algo ocurre → ENRUTADOR (uno a muchos, con filtros)
            → COLA del bus por consumidor
            → función o contenedor

y por separado
  telemetría de alto volumen → FLUJO → procesamiento y
                               almacén analítico

→ y la cola entre el enrutador y el consumidor sigue siendo
  lo que amortigua y da control de reintentos  clase 210
```

Y el error de elección más caro:

```
usar el flujo como si fuera una cola
  no hay bloqueo por mensaje ni confirmación individual
```

```
  no hay cola de fallidos integrada
  un mensaje venenoso bloquea la partición entera
  → hay que implementar todo eso a mano

y al revés: usar el bus para telemetría de alto volumen
  → caudal insuficiente y coste alto
```

2. Los mecanismos del bus

El bus tiene mecanismos que no existen en otros servicios y que resuelven problemas concretos.

```
SESIONES — orden por clave
  los mensajes con la misma clave de sesión van al mismo
  consumidor, en orden
  → resuelve «los eventos de un pedido deben procesarse en
    orden» sin serializar todo
  coste
    el caudal por sesión lo limita un solo consumidor
    y el consumidor debe mantener la sesión bloqueada
  → usar sesiones solo donde el orden importe   clase 203

TRANSACCIONES
  varias operaciones del bus en una unidad atómica
  → recibir de una cola y enviar a otra, todo o nada
  ojo   solo dentro del mismo espacio de nombres; NO
        incluye la base de datos
  → para eso sigue haciendo falta la tabla de salida
                                              clase 118

DETECCIÓN DE DUPLICADOS
  descarta mensajes con el mismo identificador dentro de
  una ventana de tiempo
  + evita duplicados del PRODUCTOR que reintenta
  − no cubre el reproceso ni la reentrega tras bloqueo
    vencido
  → NO sustituye a la idempotencia del consumidor
                                              clase 210

MENSAJES PROGRAMADOS Y DIFERIDOS
  programado   se entrega a una hora futura
  diferido     el consumidor lo aparta y lo recupera por su
               número de secuencia
  → útiles para reintentos con retroceso largo y para
    esperar a que llegue otra cosa
```

Los parámetros que producen duplicados, con la misma aritmética de la clase 210:

```
DURACIÓN DEL BLOQUEO
  el mensaje tomado queda bloqueado ese tiempo
  si el proceso tarda más, el bloqueo VENCE y el mensaje se
  reentrega
  → duplicados

  dos soluciones
    bloqueo mayor que el proceso máximo
    o RENOVAR el bloqueo desde el consumidor mientras
      trabaja
    → la renovación automática existe en las bibliotecas y
      hay que activarla

RECUENTO MÁXIMO DE ENTREGAS
  cuántas veces antes de ir a la cola de fallidos
  típico 3 a 5

TAMAÑO DE LOTE Y PREBÚSQUEDA
  la prebúsqueda trae mensajes por adelantado
  + menos latencia
  − esos mensajes están bloqueados aunque no se procesen
  → prebúsqueda alta con proceso lento = bloqueos vencidos
    y duplicados
```

Y la regla:

```
prebúsqueda × tiempo de proceso < duración del bloqueo
→ o se renueva el bloqueo activamente
```

3. El flujo y sus posiciones

El flujo funciona distinto y su operación tiene sus propias trampas.

```
PARTICIONES
  el caudal se reparte entre particiones
  la CLAVE decide en cuál cae cada evento
  → misma clave, misma partición, mismo orden  clase 116
  → y una clave caliente satura una partición mientras las
    demás están ociosas

  y el número de particiones
    condiciona el paralelismo máximo del consumo
    → más consumidores que particiones = consumidores
      ociosos

POSICIÓN DEL LECTOR
  cada grupo de consumidores guarda por dónde va
  → y esa posición hay que persistirla
  → si se pierde, se reprocesa desde el principio o se
    salta lo pendiente

  LA REGLA
    guardar la posición DESPUÉS de procesar, no antes
    → guardarla antes produce pérdida silenciosa   ley 13
    → guardarla después produce reproceso, que la
      idempotencia absorbe

RETENCIÓN
  los eventos viven un tiempo, no hasta que se consumen
  → si el consumidor está caído más que la retención, se
    pierden eventos
  → alerta por RETRASO del consumidor frente a la
    retención                                    ley 13
```

Y el mensaje venenoso, que aquí es más grave:

```
en una cola, un mensaje que falla acaba en la cola de
fallidos y los demás siguen

en un flujo, si el consumidor falla y no avanza, la
PARTICIÓN ENTERA se detiene
→ hay que decidir qué hacer con lo que no se puede
  procesar
  · apartarlo a una cola de fallidos propia y avanzar
  · o parar y alertar, si perder orden es inaceptable
→ y esa decisión hay que escribirla; no viene dada
```

Y la captura hacia almacenamiento, que ahorra trabajo:

```
el flujo puede volcar automáticamente a almacenamiento de
objetos
  → da el histórico para análisis sin escribir un
    consumidor
  → y el reproceso de meses se hace desde ahí, no desde el
    flujo                                       clase 150
```

4. Fallidos, reproceso y operación

La disciplina es la misma de la clase 210, con las piezas de Azure.

```
LA COLA DE FALLIDOS DEL BUS
  cada cola y cada suscripción tiene la suya, integrada
  los mensajes llegan con el MOTIVO y la descripción
  → y eso es lo que permite distinguir veneno de
    dependencia                                 clase 210

  y llegan también por
    caducidad del mensaje
    error de evaluación de una regla de suscripción
    y tamaño excedido
  → tres causas que no son «el consumidor falló»

LAS ALERTAS
  «hay mensajes en la cola de fallidos»          insuficiente
```

```
  «el mensaje más antiguo supera 1 hora»          ← la útil
  «la cola activa crece de forma sostenida»
  «el retraso del consumidor del flujo se acerca a la
   retención»                                     ← crítica
```

Y el reproceso, con la lección de la clase 210:

```
procedimiento escrito, con límite de ritmo y por lotes
con posibilidad de reprocesar uno solo
ejecutado por alguien que no lo escribió            ley 22
y los venenos, descartados con registro

→ reprocesar todo de golpe dispara la concurrencia y tumba
  lo que se acababa de recuperar                clase 207
```

La idempotencia, que sigue siendo obligatoria:

```
la detección de duplicados del bus cubre al productor
las sesiones cubren el orden
NINGUNA de las dos cubre
  el bloqueo vencido y la reentrega
  el reproceso desde la cola de fallidos
  el reproceso del flujo desde una posición anterior

→ clave de idempotencia del productor, registro con estados
  y escritura condicionada                      clase 210
```

Lo que hay que vigilar:

```
mensajes activos, programados y en cola de fallidos
antigüedad del más viejo, en las tres
bloqueos vencidos: señal directa de duración mal puesta
entregas repetidas del mismo mensaje
retraso del consumidor del flujo, en eventos y en tiempo
unidades de caudal del flujo frente a lo consumido
y el retraso entre que ocurre el hecho y se procesa
  → la medida que le importa al negocio         clase 211
```

Y la lista de comprobación de la clase:

- ■ el servicio elegido corresponde a lo que hay que garantizar
- ■ hay cola entre el enrutador y el consumidor
- ■ las sesiones se usan solo donde el orden importa
- ■ la duración del bloqueo supera el proceso, o se renueva
- ■ prebúsqueda × tiempo de proceso < duración del bloqueo
- ■ la detección de duplicados no se usa como sustituto de la idempotencia
- ■ toda operación con efecto es idempotente
- ■ el número de particiones del flujo permite el paralelismo necesario
- ■ la posición del lector se guarda DESPUÉS de procesar
- ■ está decidido qué hacer con un mensaje venenoso en el flujo
- ■ hay alerta por antigüedad en la cola de fallidos
- ■ hay alerta de retraso del consumidor frente a la retención
- ■ el procedimiento de reproceso existe y se ha ejecutado
- ■ los venenos se descartan con registro

Y el cierre que enlaza con la clase siguiente: con datos y mensajería en pie, falta saber qué ocurre cuando algo va mal. Observabilidad en Azure, con instrumentación estándar, es la materia de la clase 225.

Ejemplo trabajado

CloudShop monta la mensajería de su plataforma en Azure. Lo que sigue son los tres errores de elección del primer diseño, el incidente del flujo que perdió cuatro horas de telemetría, y los parámetros que quedaron.

El primer diseño, con las tres elecciones equivocadas:

```
confirmación de pedido    → enrutador de eventos, directo
                            a 3 funciones
telemetría de la web      → bus de mensajes
integración con el ERP    → enrutador de eventos
```

Y lo que falló en cada una:

```
1  CONFIRMACIÓN DE PEDIDO al enrutador, directo a funciones
   sin cola en medio
   → sin amortiguación: en campaña, las funciones escalaron
     a 2.400 y tumbaron la base            clase 207
   → sin control de reintentos por consumidor
   → y sin cola de fallidos por consumidor: los tres
     compartían destino de fallidos y no se distinguía
     cuál había fallado

   corrección
     enrutador → 3 colas del bus → 3 consumidores
     cada una con su bloqueo, reintentos y cola de fallidos

2  TELEMETRÍA de la web al bus de mensajes
   volumen                              41.000 eventos/s
   el espacio de nombres del bus se saturó
   coste estimado con el nivel necesario   3.900 €/mes

   corrección
     flujo de eventos, 8 particiones
     coste                                  340 €/mes
     y captura automática a almacenamiento para el
     histórico                            clase 150

3  INTEGRACIÓN CON EL ERP al enrutador
   el ERP exigía procesar las líneas de un pedido EN ORDEN
   el enrutador no garantiza orden
   → líneas procesadas fuera de orden: 61 pedidos con
     estado incorrecto en 3 semanas

   corrección
     cola del bus CON SESIONES, clave = identificador de
     pedido
     → orden garantizado dentro de cada pedido
     → y paralelismo entre pedidos distintos
```

El incidente del flujo: cuatro horas de telemetría perdidas.

```
situación   el consumidor del flujo se desplegó con un
            fallo y entró en bucle de reinicio
            retención del flujo                    1 día
            pero el consumidor guardaba su posición ANTES
            de procesar

qué pasó
  al arrancar, el consumidor leía un lote, guardaba la
  posición, y fallaba al procesar
  al reiniciar, leía DESDE LA POSICIÓN GUARDADA
  → los eventos de ese lote nunca se procesaron
  → y no quedaba rastro: la posición decía que estaban
    hechos

  duración del bucle                          4 h 10
  eventos perdidos                        ~610 millones
  cómo se detectó   el panel de negocio mostró 0 visitas
                    durante 4 horas             ley 15

qué salvó el caso
  la captura automática a almacenamiento seguía activa
  → los eventos estaban ahí
  → se reprocesaron desde los ficheros, en 6 h

correcciones
  1  la posición se guarda DESPUÉS de procesar
     → produce reproceso, que la idempotencia absorbe
  2  alerta de retraso del consumidor: «el retraso supera
     el 50 % de la retención»
  3  alerta de reinicios del consumidor       clase 221
  4  y una comprobación de negocio: «visitas por minuto
     por debajo del mínimo esperado»
     → esta es la que lo habría detectado en 5 minutos
                                              clase 211
```

Los duplicados del bus, y el bloqueo.

```
síntoma   bajo carga, algunos pedidos generaban dos
          notificaciones

diagnóstico
  duración del bloqueo                      30 s (defecto)
  prebúsqueda                               20 mensajes
  tiempo de proceso por mensaje             1,8 s

  20 × 1,8 = 36 s > 30 s
  → los últimos mensajes del lote perdían el bloqueo antes
    de procesarse
  → se reentregaban a otro consumidor

  y la métrica que lo confirmaba
    «bloqueos vencidos»: 340/hora en el pico
    → estaba disponible y no estaba en ningún panel
                                                    ley 15

correcciones
  duración del bloqueo                      30 s → 120 s
  prebúsqueda                               20 → 10
  renovación automática del bloqueo         activada
  e idempotencia en el consumidor de notificaciones

bloqueos vencidos                          340/h → 0
notificaciones duplicadas                    61 → 0
```

La configuración final:

```
ENRUTADOR DE EVENTOS
  eventos de negocio y de la plataforma
  filtros por tipo y por contenido
  destino: colas del bus, nunca funciones directas
  reintentos: 5 con retroceso; destino de fallidos
  configurado

BUS DE MENSAJES
  cola             bloqueo  prebúsq.  entregas  sesiones
  notificaciones    120 s     10         5        no
  inventario        180 s     10         5        no
  facturación       300 s      5         5        no
  erp               240 s      1         3        SÍ
  → la de sesiones con prebúsqueda 1: el orden importa más
    que la latencia

  detección de duplicados activada en la cola de pedidos,
  ventana de 10 minutos
  → y aun así, idempotencia en los cuatro consumidores

FLUJO
  8 particiones, clave = identificador de sesión de usuario
  2 grupos de consumidores: proceso en tiempo real y
    análisis
  retención 3 días (subida desde 1)
  captura automática a almacenamiento, cada 5 min
  posición guardada después de procesar
```

Las alertas montadas:

```
antigüedad del mensaje más viejo en cada cola de fallidos
  umbral 1 hora → canal de guardia
cola activa creciendo de forma sostenida
bloqueos vencidos > 0
retraso del consumidor del flujo > 50 % de la retención
eventos por minuto por debajo del mínimo esperado
  → la de negocio, que detecta lo que las técnicas no
reinicios de consumidores
```

El reproceso, probado:

```
primer ensayo
  se dejaron 8.000 mensajes en la cola de fallidos a
  propósito
```

```
  quien lo ejecutó no había escrito el procedimiento
  → 2 de 7 pasos necesitaron aclaración
  → y el límite de ritmo estaba en el paso 4, no en el 1
    → se movió al 1                              clase 216

segundo ensayo
  8.000 mensajes reprocesados en 3 min
  concurrencia máxima                            40
  incidencias                                     0
```

El resultado:

```
notificaciones duplicadas                      61          0
pedidos con estado incorrecto (orden)          61          0
eventos de telemetría perdidos            610 M            0
coste de la telemetría                    3.900 €      340 €
bloqueos vencidos                         340/hora        0
tiempo de detección de un consumidor
  parado                                     4 h 10      5 min
reprocesos ejecutados                           0          4
  que causaron incidente                        —          0
```

La lección que esta clase deja: los tres errores del primer diseño fueron de elección de servicio, no de configuración: el enrutador directo a funciones sin amortiguar, el bus para telemetría de gran volumen y el enrutador para algo que necesitaba orden. Y el incidente más grave —seiscientos diez millones de eventos perdidos— lo causó guardar la posición del lector antes de procesar, un detalle de una línea que producía pérdida silenciosa; lo salvó una captura automática a almacenamiento que se había activado por otro motivo.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-18-azure-production-architecture/224-service-bus-event-grid-y-event-hubs/lab.py
```

El laboratorio selecciona el motor de práctica messaging 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 azure-event-platform 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 flujo que declara orden, entrega y manejo de errores. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye azure-event-platform 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 pico de eventos tumba la base de datos	El enrutador entrega directamente a las funciones, sin cola que amortigüe	Pon una cola por consumidor entre el enrutador y el proceso, con su bloqueo, reintentos y cola de fallidos propios.
Se procesan mensajes fuera de orden y el estado queda mal	Se eligió un servicio que no garantiza orden	Usa colas con sesiones y clave por entidad; el orden se garantiza dentro de la sesión y sigue habiendo paralelismo entre entidades.
Aparecen duplicados solo con carga alta	La prebúsqueda por el tiempo de proceso supera la duración del bloqueo	Aumenta el bloqueo, reduce la prebúsqueda y activa la renovación automática; vigila la métrica de bloqueos vencidos.
Se pierden eventos sin ningún error	La posición del lector se guarda antes de procesar	Guarda la posición después de procesar y absorbe el reproceso con idempotencia; alerta si el retraso se acerca a la retención.
Un mensaje incorrecto detiene todo el consumo	En un flujo, el consumidor no avanza y bloquea la partición entera	Decide y escribe qué hacer con el veneno: apartarlo a una cola propia y avanzar, o parar y alertar.
La factura de mensajería es muy alta para telemetría	Se usa el bus de mensajes para volúmenes que corresponden a un flujo	Usa el flujo para gran volumen, con captura automática a almacenamiento para el histórico.

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 garantiza el bus que no garantiza el enrutador de eventos?
2. ¿Qué relación deben cumplir prebúsqueda, tiempo de proceso y duración del bloqueo?
3. ¿Por qué la detección de duplicados no sustituye a la idempotencia?
4. ¿Cuándo se debe guardar la posición del lector de un flujo y por qué?
5. ¿Qué ocurre con un mensaje venenoso en un flujo y qué hay que decidir?

Referencias

- Microsoft (2025). Azure Service Bus: message sessions, transactions and dead-letter queues. <https://learn.microsoft.com/en-us/azure/service-bus-messaging/service-bus-messaging-overview>
- Microsoft (2025). Azure Event Grid: event delivery and retry. <https://learn.microsoft.com/en-us/azure/event-grid/delivery-and-retry>
- Microsoft (2025). Azure Event Hubs: partitions, consumer groups and capture. <https://learn.microsoft.com/en-us/azure/event-hubs/event-hubs-features>
- Microsoft (2025). Choose between Azure messaging services. <https://learn.microsoft.com/en-us/azure/service-bus-messaging/compare-messaging-services>
- Hohpe, G. y Woolf, B. (2003). Enterprise Integration Patterns. <https://www.enterpriseintegrationpatterns.com/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 18 en PDF · Recorrido de Azure en PDF · Manual integral

Anterior	Índice	Siguiente
← 223 · Azure SQL, Cosmos DB y consistencia distribuida	Parte 18 · Programa	225 · Azure Monitor, Application Insights y OpenTelemetry →

Evaluación — 224 Service Bus, Event Grid y Event Hubs

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 azure-event-platform 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.

225 — Azure Monitor, Application Insights y OpenTelemetry

Parte: 18 — Azure: arquitectura empresarial y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: observability · Estado: EXECUTABLECORE

Propósito

Montar la observabilidad de Azure con instrumentación estándar, de forma que no haya que rehacerla al cambiar de nube ni de proveedor de análisis. La clase cubre el destino único de registros y métricas, el muestreo que decide el coste, y los dos asuntos que este programa lleva señalando y que aquí tienen forma propia: la ingesta sin límite factura más que el cómputo, y la instrumentación específica de un proveedor se convierte en un cambio caro el día que se quiere mover.

Resultados de aprendizaje

Al finalizar podrás:

1. Enrutar registros y métricas de todos los recursos a un destino común.
2. Instrumentar con el estándar abierto para no quedar atado.
3. Controlar el coste con muestreo, niveles y planes de tabla.
4. Definir alertas y objetivos junto al servicio, como código.
5. Correlacionar desde la alerta hasta la causa sin eslabones rotos.

Conceptos centrales

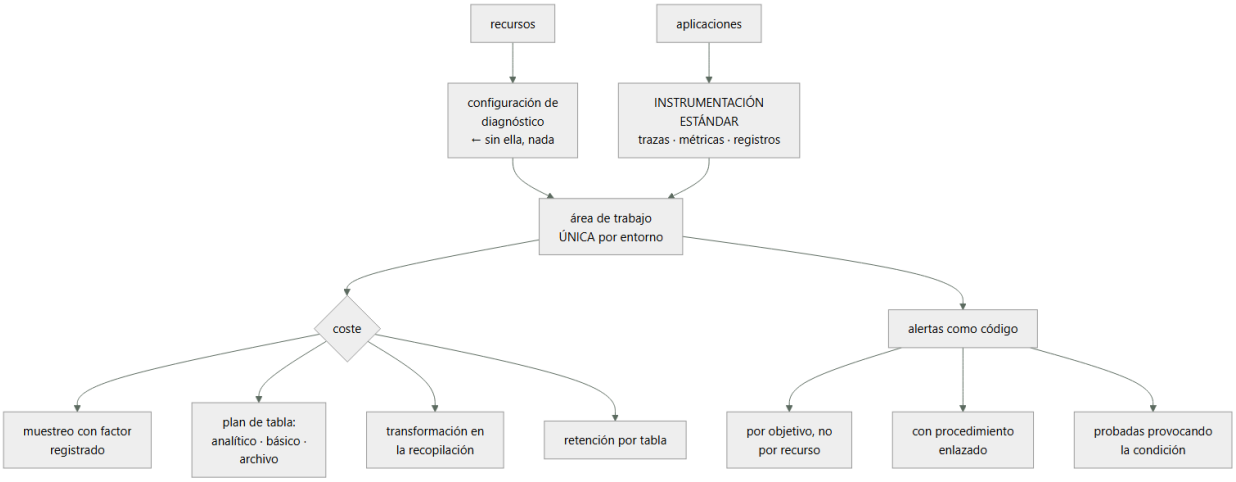
Concepto	Comprensión verificable
área de trabajo	Destino de registros y métricas, con su retención, sus permisos y su facturación.
configuración de diagnóstico	Ajuste por recurso que decide qué categorías se envían y a dónde. Sin ella, el recurso no registra nada.
muestreo adaptativo	Reducción automática del volumen enviado, con factor registrado para poder extrapolar.
instrumentación estándar	Emisión de trazas, métricas y registros con el estándar abierto, independiente del destino.
plan de tabla	Nivel de una tabla de registros: analítico, básico o de archivo. Cambia precio y capacidad de consulta.
regla de recopilación	Definición de qué datos se recogen de qué recursos y con qué transformación antes de almacenarlos.

Modelo mental

La arquitectura Azure nace del tenant y las políticas heredables, y llega al workload mediante identidades administradas y redes privadas.

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

■ Flujo de razonamiento



Desarrollo

1. Un destino, y la configuración que casi nadie pone

En Azure, un recurso no envía nada por defecto. La configuración de diagnóstico es lo que enruta sus registros y métricas, y si falta, ese recurso es invisible.

```
LO QUE HAY QUE DECIDIR POR RECURSO
  qué categorías de registro se envían
  si se envían métricas
  y a qué destino: área de trabajo, almacenamiento, flujo

Y LO QUE PASA SI FALTA
  el recurso funciona
  no aparece en ninguna consulta
  y cuando hay un incidente, no hay datos de las horas
  anteriores                                          ley 13
```

Y por eso se resuelve con política, no con disciplina:

```
política de tipo «desplegar si no existe» que crea la
configuración de diagnóstico en cada recurso nuevo
                                                  clase 217
→ y con la estimación de coste hecha ANTES de asignarla
→ y con las categorías elegidas, no todas
```

El destino, con la decisión de cuántas áreas de trabajo:

```
UNA POR ENTORNO, no una por suscripción
  → correlacionar entre servicios exige que estén juntos
  → y las consultas cruzadas entre áreas son más lentas y
    más incómodas

Y LOS PERMISOS SE RESUELVEN APARTE
  quien no debe ver los registros de otro equipo se
  controla con permisos por tabla o por recurso
  → no separando áreas                          clase 218

EXCEPCIONES RAZONABLES
  registros de auditoría y de seguridad, en un área
  aparte, en una suscripción donde el comprometido no
  pueda borrarlos                          clases 141, 226
```

Y los planes de tabla, que son la palanca de coste más directa:

```
ANALÍTICO   consulta completa, alertas, retención larga
            el más caro por gigabyte
BÁSICO      consulta limitada, sin alertas, retención corta
            mucho más barato
            → para registros voluminosos que solo se miran
              al diagnosticar
ARCHIVO     almacenamiento barato, restauración bajo demanda
            → para lo que hay que conservar por norma
```

```
→ poner en básico las tablas de registros voluminosos de
  aplicación suele reducir la factura a la mitad
→ y hay que comprobar antes que no se necesitan alertas
  sobre ellas
```

2. Instrumentar con el estándar abierto

Aquí hay una decisión de arquitectura que se toma sin pensarla y se paga años después.

```
INSTRUMENTACIÓN ESPECÍFICA DEL PROVEEDOR
  la biblioteca del proveedor, con su API
  + integración inmediata y algunas funciones extra
  − el código queda atado: cambiar de destino exige tocar
    todas las aplicaciones                    clase 158

INSTRUMENTACIÓN ESTÁNDAR (OpenTelemetry)
  API y formato comunes; el destino es configuración
  + cambiar de proveedor de análisis es un cambio de
    exportador, no de código
  + la misma instrumentación vale en las tres nubes
  − algunas funciones del proveedor no están cubiertas

→ y esta es la decisión: el código emite en estándar y el
  destino se configura
→ el coste de cambiarla más adelante es alto: toca todas
  las aplicaciones                              ley 14
```

Y lo que hay que emitir, con la disciplina de la clase 211:

```
TRAZAS con contexto propagado
  incluidas las llamadas asíncronas: el contexto viaja
  DENTRO del mensaje                          clase 224
  → sin eso, la traza se corta en la primera cola

MÉTRICAS
  las cuatro señales y las de negocio
  con dimensiones de baja cardinalidad        clase 211
  → el identificador de pedido va al registro, no a una
    dimensión

REGISTROS estructurados
  con traza, servicio, versión, entorno y duración
  y sin secretos ni datos personales completos
```

Y el detalle que decide la utilidad del conjunto:

```
EL MISMO IDENTIFICADOR DE TRAZA en las tres señales
  → así se salta de una métrica a las trazas del periodo,
    y de una traza a sus registros
  → si cada capa usa su propio identificador, la cadena se
    rompe                                     clase 211
```

El muestreo, que es lo que hace viable el coste:

```
MUESTREO ADAPTATIVO
  el agente reduce el volumen según la carga y registra el
  FACTOR aplicado
  → los recuentos se pueden extrapolar
  → y por eso hay que usar las funciones que lo tienen en
    cuenta al contar

QUÉ SE MUESTREA Y QUÉ NO
  peticiones correctas        1-5 %
  errores                     100 %, siempre
  peticiones lentas           100 %
  y todo lo de un usuario concreto, activable para
    diagnosticar

EL ERROR HABITUAL
  muestrear y luego contar sin tener en cuenta el factor
  → los paneles muestran cifras que no son
```

3. Alertas y objetivos, como código

La disciplina es la de la clase 211, con las piezas de esta nube.

```
LO QUE SE DECLARA JUNTO AL SERVICIO
  configuración de diagnóstico
  reglas de recopilación
  consultas guardadas de diagnóstico
  panel del servicio
  objetivo de nivel de servicio
  reglas de alerta con su grupo de acción

→ y una función de aptitud: ningún servicio se despliega
  sin ellas                                    clase 190
```

Los tipos de alerta, con el uso de cada uno:

```
POR MÉTRICA        rápida y barata; para señales continuas
POR CONSULTA       flexible; para lo que exige agregar o
                   correlacionar
                   → con coste por ejecución: cuidado con
                     la frecuencia
POR ACTIVIDAD      cambios en recursos, políticas y
                   permisos
                   → «alguien quitó una asignación de
                     política»                 clase 217
POR SALUD DEL
  SERVICIO         incidencias del propio proveedor
                   → y esta la olvida casi todo el mundo
```

Y las que este programa exige siempre:

```
ALERTA POR AUSENCIA
  «esta función no se ha ejecutado en N minutos»
  → un trabajo programado que deja de dispararse no genera
    error                                        ley 13

ALERTA POR ANTIGÜEDAD
  cola de fallidos, certificados, sincronización
                                  clases 224, 196, 222

Y POR RITMO DE CONSUMO DEL PRESUPUESTO DE ERROR
  → la que despierta, en vez del error suelto  clase 211
```

Y el destino, con la lección repetida:

```
un grupo de acción por turno de guardia, no uno por
servicio
→ crear un grupo por equipo garantiza que alguno quede sin
  destinatarios                        ley 15, clases 210, 211

y la comprobación
  provocar la condición y ver si llega              ley 22
  → en la clase 211, 7 de 40 no llegaron a la primera
```

Y una nota sobre el coste de las alertas por consulta:

```
una alerta por consulta que se ejecuta cada minuto sobre
mucho volumen factura
→ frecuencia acorde a lo que se detecta
→ y una alerta que nunca se ha disparado en un año, o
  sobra o su umbral está mal                   clase 190
```

4. Consultar, correlacionar y controlar el gasto

Las consultas preparadas, que ahorran minutos durante un incidente:

```
«dame todo lo de esta traza, en las tres señales»
«errores de este servicio en la última hora, por tipo»
«peticiones más lentas y qué dependencia domina»
«qué cambió en los recursos en la última hora»
«compara esta hora con la misma de ayer»

→ guardadas y enlazadas desde el procedimiento de la
  alerta                                       clase 127
```

La cadena de diagnóstico, con sus eslabones:

```
alerta por objetivo
  → panel del servicio
    → trazas del periodo
```

```
      → registros de esas trazas
        → registro de actividad: ¿qué se desplegó o cambió?
                                                clase 122

→ y si un eslabón falta, el diagnóstico se hace a ojo
```

El control del gasto, que aquí tiene palancas concretas:

```
1  CATEGORÍAS ELEGIDAS, no todas
   muchas categorías de diagnóstico son voluminosas y
   nadie las consulta

2  TRANSFORMACIÓN EN LA RECOPILACIÓN
   filtrar y recortar antes de almacenar
   → quitar campos que no se usan, descartar líneas de
     nivel bajo
   → es lo que más reduce sin perder capacidad de
     diagnóstico

3  PLAN DE TABLA por volumen y uso

4  RETENCIÓN por tabla, no global
   → los registros de auditoría, largos; los de aplicación,
     cortos                                    clase 141

5  MUESTREO en la aplicación

6  Y COMPROMISO DE CAPACIDAD si el volumen es estable
   → descuento por gigabyte comprometido
```

Y la comprobación honesta:

```
¿cuánto cuesta la observabilidad frente al cómputo?
→ en la clase 211 costaba el doble
→ y en la 214 era el 11 % de la factura total
→ conviene mirarlo cada trimestre, no una vez  clase 214
```

Y la lista de comprobación de la clase:

- ■ hay política que crea la configuración de diagnóstico en cada recurso
- ■ las categorías están elegidas, no todas
- ■ hay un área de trabajo por entorno, no por suscripción
- ■ los registros de auditoría van a un área separada e inalcanzable desde producción
- ■ la instrumentación de las aplicaciones es estándar
- ■ el contexto de traza viaja dentro de los mensajes
- ■ las dimensiones no tienen cardinalidad alta
- ■ el muestreo registra el factor y las consultas lo tienen en cuenta
- ■ los errores y las peticiones lentas no se muestrean
- ■ paneles, objetivos y alertas están en la plantilla del servicio
- ■ hay alertas por ausencia, por antigüedad y por ritmo de consumo
- ■ hay alerta de salud del servicio del proveedor
- ■ el destino es un grupo de acción con guardia
- ■ cada alerta se ha probado provocando la condición
- ■ hay transformación en la recopilación y planes de tabla por volumen
- ■ el coste de observabilidad se revisa cada trimestre

Y el cierre que enlaza con la clase siguiente: con la observabilidad en pie, queda la parte que mira lo mismo con otra intención: detectar lo que no debería estar pasando. Postura de seguridad, cumplimiento y detección es la materia de la clase 226.

Ejemplo trabajado

CloudShop monta la observabilidad de su plataforma en Azure. Lo que sigue son los recursos que no registraban nada, la factura de ingesta que superaba al cómputo, y la decisión de instrumentación que evitó un cambio de meses dos años después.

El punto de partida:

```
recursos                                        14.200
  con configuración de diagnóstico                 940   6,6 %
  sin ella                                      13.260

áreas de trabajo                                    31
  una por suscripción, creadas al azar
  → correlacionar entre servicios exigía consultas
    cruzadas entre 4 o 5 áreas

instrumentación de aplicaciones
  con la biblioteca específica del proveedor        14
  con estándar abierto                               0

coste mensual de observabilidad              6.900 €
  ingesta                                    4.400 €
  retención                                  1.800 €
  alertas por consulta                         700 €
```

Y lo que reveló el primer incidente serio:

```
un fallo del cortafuegos del centro dejó sin conectividad
a 9 radios durante 40 minutos

al investigar
  el cortafuegos NO tenía configuración de diagnóstico
  → no había ni un registro de las horas anteriores
  → la causa se dedujo por descarte, en 3 horas

y los 13.260 recursos sin diagnóstico incluían
  el cortafuegos central
  4 de las 6 pasarelas
  todas las cuentas de almacenamiento
  y las 43 redes virtuales
```

Las correcciones estructurales:

```
POLÍTICA que crea la configuración de diagnóstico en cada
recurso nuevo y remedia los existentes         clase 217

  y la estimación previa del coste
    si se enviaban TODAS las categorías        ~9.400 €/mes
    con categorías elegidas                    ~1.900 €/mes

  categorías elegidas
    cortafuegos     reglas aplicadas, DNS, amenazas
                    → no: estadísticas detalladas por flujo
    almacenamiento  operaciones de escritura y borrado
                    → no: cada lectura
    redes           registros de flujo, ya existentes
    bases           auditoría, consultas lentas
                    → no: cada consulta

ÁREAS DE TRABAJO
  31 → 3 (producción, preproducción, desarrollo)
  + 1 de auditoría y seguridad, en la suscripción de
    seguridad, sin acceso desde producción     clase 141
  permisos por tabla para separar lo que cada equipo ve
```

La instrumentación: la decisión que se registró.

```
el equipo iba a seguir con la biblioteca específica
«funciona y es lo que sabemos»

el argumento que cambió la decisión
  la parte 19 iba a montar lo mismo en otra nube
  y el equipo de datos ya usaba otro proveedor de análisis
  para sus cargas
  → tres destinos distintos, y el código atado a uno

decisión
  migrar a instrumentación estándar
  el destino es un exportador configurado, no código
  coste de la migración              14 aplicaciones,
                                     6 semanas
```

```
registro de decisión                          clase 190
  premisa   habrá al menos dos destinos de análisis
  qué la reabriría   si el proveedor deja de soportar el
                     estándar, o si su coste sube por
                     encima del propio

y dos años después
  se cambió el proveedor de análisis de las cargas de datos
  cambio necesario en las aplicaciones            ninguno
  se cambió el exportador y el destino
  tiempo                                           2 días
  → sin esta decisión, habrían sido meses de tocar 14
    aplicaciones                                 clase 158
```

El coste, atacado con las cinco palancas:

```
ingesta inicial tras aplicar las políticas    1.900 €/mes

1  TRANSFORMACIÓN EN LA RECOPILACIÓN
   los registros de aplicación traían 41 campos; se
   consultaban 9
   se descartan las líneas de nivel de información de 3
   servicios muy habladores
   1.900 € → 1.140 €

2  PLANES DE TABLA
   registros de flujo de red y de contenedores → básico
   → no se necesitaban alertas sobre ellos, comprobado
   1.140 € → 640 €

3  RETENCIÓN POR TABLA
   aplicación         30 días
   flujo de red       14 días
   auditoría          400 días, en archivo tras 90
   retención                       1.800 € → 410 €

4  MUESTREO EN LA APLICACIÓN
   peticiones correctas al 4 %, errores y lentas al 100 %
   → y las consultas ajustadas para tener en cuenta el
     factor
   → aquí hubo un susto: durante dos semanas, el panel de
     pedidos por minuto mostró cifras 25 veces menores
   → porque la consulta contaba filas sin aplicar el
     factor de muestreo

5  ALERTAS POR CONSULTA
   41 alertas por consulta, 12 de ellas cada minuto
   → se bajó la frecuencia de las que detectan cosas
     lentas
   700 € → 210 €

total                              6.900 € → 1.260 €/mes
```

Las alertas, revisadas:

```
antes                                            112
  disparadas al mes                              890
  accionables                                 71 (8 %)
  con destino a grupos de acción por equipo        63
    de ellos, SIN destinatarios                    14

después                                           58
  por objetivo de nivel de servicio                 9
  por ausencia                                     11
    · funciones y trabajos programados
  por antigüedad                                    8
    · colas de fallidos, certificados, sincronización
  por actividad                                     6
    · cambios de política, de permisos, de rutas
  por salud del servicio del proveedor              1
    → esta detectó 2 incidencias del proveedor antes de
      que nadie las notara
  el resto, técnicas                               23
```

```
  destino   un grupo de acción por turno
  disparadas al mes                                74
  accionables                                 68 (92 %)

y la prueba de extremo a extremo
  58 alertas, condición provocada una a una
  no llegaron                                       6
    3 por umbral mal puesto
    2 por destino equivocado
    1 porque la consulta tenía un error de sintaxis y la
      regla estaba en estado de error desde su creación
                                                  ley 22
```

La cadena de diagnóstico, comprobada:

```
02:41  alerta: el objetivo de confirmación de pedido
       consume presupuesto de error 11× más rápido
02:41  el mensaje enlaza el panel del servicio
02:42  el panel muestra el p99 disparado desde las 02:33
02:42  registro de actividad: cambio en una regla del
       cortafuegos a las 02:31                clase 122
02:43  trazas del periodo: el tramo lento es la llamada al
       servicio de precios
02:44  registros de esas trazas: tiempo de espera agotado
       al conectar
02:46  se revierte el cambio del cortafuegos
02:51  restablecido

tiempo hasta identificar la causa                 3 min
```

El resultado:

```
recursos con diagnóstico                    6,6 %      99,1 %
áreas de trabajo                              31           4
coste de observabilidad                   6.900 €     1.260 €
alertas configuradas                          112          58
  accionables                             71 (8 %)   68 (92 %)
alertas sin destinatarios                     14           0
instrumentación estándar                    0/14       14/14
tiempo de cambio de proveedor de análisis   meses      2 días
tiempo medio hasta identificar causa      3 h (est.)     3 min
```

La lección que esta clase deja: el 93 % de los recursos no registraba nada, incluido el cortafuegos central, y eso se descubrió en un incidente en el que no había datos de las horas anteriores. La factura de observabilidad superaba a la de cómputo y se redujo un 82 % sin perder capacidad de diagnóstico, con cinco palancas de configuración. Y la decisión que más valió no dio ningún beneficio el primer año: instrumentar con el estándar abierto, que dos años después convirtió un cambio de proveedor de meses en dos días.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-18-azure-production-architecture/225-azure-monitor-application-insights-y-opentelemetry/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 azure-observability 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 azure-observability 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 recurso no aparece en ninguna consulta durante un incidente	No tiene configuración de diagnóstico y por defecto no envía nada	Crea la configuración con una política que la despliegue en todo recurso nuevo, con categorías elegidas y coste estimado antes.
Correlacionar entre servicios exige consultas cruzadas entre muchas áreas	Se creó un área de trabajo por suscripción	Una por entorno, con permisos por tabla para separar lo que cada equipo ve, y una aparte para auditoría.
Cambiar de proveedor de análisis obligaría a tocar todas las aplicaciones	Se instrumentó con la biblioteca específica del proveedor	Emite con el estándar abierto y deja el destino como configuración; el coste de cambiarlo después es alto.
Los paneles muestran cifras muy inferiores a la realidad	Hay muestreo y las consultas cuentan filas sin aplicar el factor	Usa las funciones que tienen en cuenta el factor de muestreo y comprueba los recuentos contra una fuente independiente.
La factura de observabilidad supera a la de cómputo	Todas las categorías, sin transformación, en plan analítico y con retención larga	Elige categorías, transforma en la recopilación, usa planes de tabla por volumen y fija retención por tabla.
Una alerta nunca ha llegado a nadie	Su grupo de acción no tiene destinatarios o la regla está en estado de error	Un grupo de acción por turno de guardia y prueba de extremo a extremo provocando la condición de cada alerta.

■ 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 ocurre con un recurso que no tiene configuración de diagnóstico?
2. ¿Por qué conviene un área de trabajo por entorno y no por suscripción?
3. ¿Qué se gana instrumentando con el estándar abierto y cuándo se cobra esa ganancia?
4. ¿Qué error de consulta produce el muestreo si no se tiene en cuenta?

5. ¿Cuáles son las cinco palancas para reducir el coste de ingesta?

Referencias

- Microsoft (2025). Azure Monitor: diagnostic settings. <https://learn.microsoft.com/en-us/azure/azure-monitor/essentials/diagnostic-settings>
- Microsoft (2025). Application Insights with OpenTelemetry. <https://learn.microsoft.com/en-us/azure/azure-monitor/app/opentelemetry-enable>
- Microsoft (2025). Log Analytics table plans and data transformations. <https://learn.microsoft.com/en-us/azure/azure-monitor/logs/data-platform-logs>
- Microsoft (2025). Sampling in Application Insights. <https://learn.microsoft.com/en-us/azure/azure-monitor/app/sampling>
- OpenTelemetry (2025). Specification and semantic conventions. <https://opentelemetry.io/docs/specs/otel/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 18 en PDF · Recorrido de Azure en PDF · Manual integral

Anterior	Índice	Siguiente
← 224 · Service Bus, Event Grid y Event Hubs	Parte 18 · Programa	226 · Defender for Cloud, Policy y Sentinel →

Evaluación — 225 Azure Monitor, Application Insights y OpenTelemetry

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 azure-observability 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.

226 — Defender for Cloud, Policy y Sentinel

Parte: 18 — Azure: arquitectura empresarial y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: security · Estado: EXECUTABLECORE

Propósito

Montar la seguridad operativa de Azure: la evaluación continua de la postura, el cumplimiento medido contra un marco, y la detección con su parte incómoda. La clase separa lo que cada herramienta hace de verdad, ordena las recomendaciones por alcance en vez de por su puntuación, y desarrolla el asunto que decide si un centro de operaciones funciona: una detección que nadie ha comprobado no detecta, y una que genera mil alertas al día no se mira.

Resultados de aprendizaje

Al finalizar podrás:

1. Distinguir postura, cumplimiento y detección, y qué resuelve cada uno.
2. Priorizar recomendaciones por alcance real, no por la puntuación.
3. Medir el cumplimiento contra un marco sin convertirlo en trámite.
4. Configurar detección con reglas propias y comprobarlas.
5. Controlar el coste de la ingesta de seguridad.

Conceptos centrales

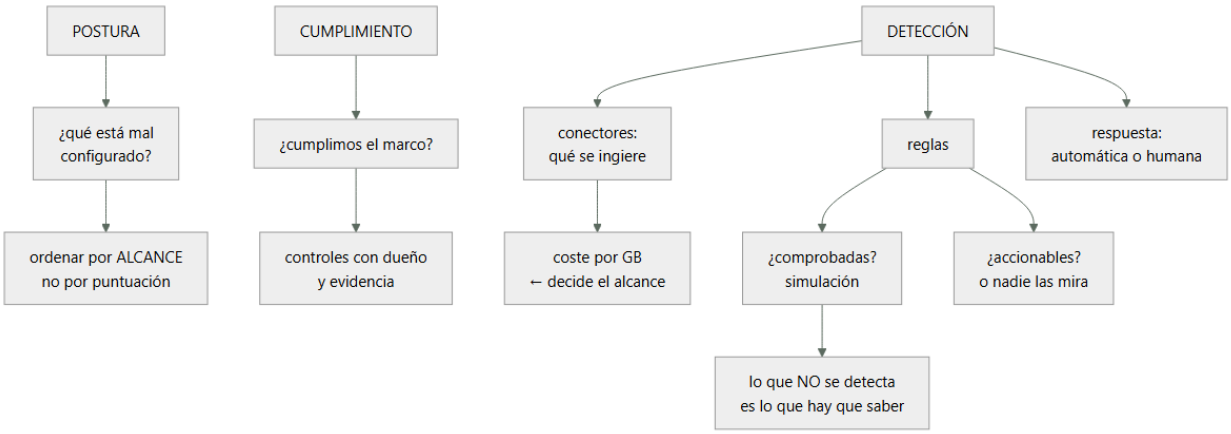
Concepto	Comprensión verificable
postura de seguridad	Evaluación continua de la configuración frente a buenas prácticas. Dice qué está mal configurado.
puntuación de seguridad	Cifra agregada de la postura. Útil como tendencia, engañosa como prioridad.
cumplimiento normativo	Medida de la configuración contra un marco concreto, con sus controles.
detección	Reglas que generan incidentes a partir de señales. Requiere ingesta, reglas y quien las atienda.
simulación de ataque	Ejecución controlada de una técnica conocida para comprobar si se detecta.
conector de datos	Origen de señales que se ingiere para detectar. Cada uno tiene su volumen y su coste.

Modelo mental

La arquitectura Azure nace del tenant y las políticas heredables, y llega al workload mediante identidades administradas y redes privadas.

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. Postura, cumplimiento y detección

Tres cosas distintas que se mezclan y responden preguntas diferentes.

```
POSTURA
  ¿qué está mal configurado, ahora mismo?
  evaluación continua contra buenas prácticas
  produce   recomendaciones
  ejemplo   «esta base tiene acceso público»

CUMPLIMIENTO
  ¿cumplimos este marco concreto?
  la misma información, agrupada por controles de una norma
  produce   evidencia para auditoría
  ejemplo   «el control A.9.4 está cubierto por estas 12
             políticas»

DETECCIÓN
  ¿está ocurriendo algo ahora?
  reglas sobre señales que generan incidentes
  produce   incidentes que alguien atiende
  ejemplo   «esta identidad ha accedido desde dos países en
             10 minutos»
```

Y la relación entre las tres, que ordena el trabajo:

```
la POSTURA reduce lo que hay que detectar
  → un almacén sin acceso público no puede ser accedido
    desde internet
  → arreglar configuración es más barato que detectar su
    explotación

el CUMPLIMIENTO no mejora la seguridad por sí mismo
  → mide y da evidencia; no cierra nada
  → y perseguir el porcentaje produce trámite   ley 17

la DETECCIÓN cubre lo que la configuración no puede
  → uso legítimo de credenciales robadas, abuso de
    permisos concedidos, comportamiento anómalo
```

La puntuación de seguridad, con su trampa:

```
es una cifra agregada y ponderada
  + útil como TENDENCIA: si baja, algo se ha degradado
  – engañosa como prioridad: la recomendación que más
    sube la puntuación no es la que más riesgo reduce

  ejemplo típico
    «activar cifrado en 400 discos de desarrollo»
      +6 puntos, riesgo bajo
    «una identidad con Propietario en producción»
      +0,2 puntos, riesgo máximo         clase 218

→ ordena por ALCANCE desde el punto comprometido, no por
```

```
  puntos                                          clase 189
```

2. Postura: priorizar bien

La herramienta produce cientos de recomendaciones. El trabajo es ordenarlas.

```
LAS PREGUNTAS QUE ORDENAN
  1  ¿está expuesto a internet?
  2  ¿tiene datos sensibles?
  3  ¿hasta dónde se llega desde ahí?     ← la que decide
  4  ¿existe ya explotación conocida de esa vulnerabilidad?
  5  ¿cuántos recursos afecta?

→ y las herramientas modernas calculan CAMINOS DE ATAQUE:
  «desde esta máquina expuesta se llega, por esta
   identidad, a esta base con datos de clientes»
→ eso vale más que cualquier lista ordenada por puntos
```

Y las categorías de recomendación, con lo que cuesta cada una:

```
CONFIGURACIÓN (acceso público, cifrado, versión de TLS)
  → se resuelven con política, en masa           clase 217
  → y lo nuevo se impide con denegación

IDENTIDAD (permisos amplios, sin segundo factor)
  → las de más alcance y las más difíciles de arreglar
                                                 clase 218

VULNERABILIDADES (software sin parchear)
  → parcheo y actualización de imágenes          clase 254
  → y con exploración continua, no solo al construir
                                                 clase 212

DATOS (almacenes sin clasificar, sin cifrado propio)
  → hay que saber dónde están los datos sensibles primero
```

Y el orden de trabajo que funciona:

```
1  cerrar lo expuesto a internet con datos detrás
2  reducir el alcance de las identidades más amplias
3  impedir por política que lo arreglado vuelva a ocurrir
4  remediar el resto por lotes, con tareas automáticas
5  y aceptar por escrito lo que no compensa arreglar
                                                 clase 189
```

Y una advertencia sobre las exenciones:

```
una recomendación se puede marcar como exenta
→ y sin dueño ni fecha, desaparece del radar para siempre
                                                    ley 25
→ toda exención con motivo, nombre y caducidad, y un panel
  de las que vencen                              clase 217
```

El cumplimiento, con la disciplina que lo hace útil:

```
se elige el marco que aplica de verdad, no todos
cada control tiene DUEÑO y evidencia
lo que la herramienta no puede comprobar se documenta
  aparte
  → una parte importante de cualquier marco es
    organizativa, no técnica

y la medida honesta
  no «cumplimos el 94 %»
  sino «de los 114 controles, 71 automatizados y verdes,
  22 automatizados con excepciones, 21 manuales con
  evidencia, y estos 3 no los cumplimos, con este plan»
```

3. Detección: lo que cuesta y lo que hay que comprobar

Un centro de operaciones de seguridad tiene tres partes y las tres tienen que existir.

```
1  INGESTA        qué señales entran
2  REGLAS         qué se considera sospechoso
3  RESPUESTA      quién actúa, y cómo

→ si falta cualquiera, no hay detección: hay una consola
```

La ingesta, que decide el coste y el alcance:

```
CONECTORES HABITUALES, por valor y por volumen
  registros de identidad y de inicio de sesión
    valor ALTO, volumen medio      ← el primero, siempre
  registro de actividad de la nube
    valor alto, volumen bajo       ← barato y útil
  alertas de la propia herramienta de postura
    valor alto, volumen muy bajo
  registros de dispositivos y puntos finales
    valor alto, volumen alto
  registros de red (flujos, cortafuegos)
    valor medio, volumen MUY alto  ← el que dispara la
                                     factura
  registros de aplicación
    valor variable

→ y la decisión no es «todo»: es qué se ingiere para
  detectar, qué se guarda barato para investigar y qué no
  se guarda                                    clase 225
```

Y el patrón que resuelve el coste:

```
lo que alimenta REGLAS      → plan analítico
lo que solo se consulta al
  investigar                → plan básico o archivo
lo que hay que conservar
  por norma                 → archivo

→ y las reglas de detección sobre datos en plan básico no
  se pueden hacer: por eso hay que decidir antes
```

Las reglas, con las dos preguntas que deciden si valen:

```
¿ESTÁ COMPROBADA?
  simular la técnica que dice detectar
  → en la clase 179, 2 de 6 técnicas simuladas NO se
    detectaban, y las reglas existían            ley 22
  → y en la clase 189, 3 de 14 pruebas fallaron sobre
    controles documentados como implantados

¿ES ACCIONABLE?
  ¿qué hace quien la recibe?
  → si no hay respuesta posible, la regla genera ruido
  → y el ruido esconde lo real                 clase 125
```

Y la calibración, que es trabajo continuo:

```
una regla nueva se despliega en modo de observación
se mide cuántas veces se dispararía y sobre qué
se ajustan las exclusiones legítimas
  → el escáner de vulnerabilidades propio
  → la cuenta de la copia de seguridad
  → los rangos de la sede
y solo entonces genera incidentes

→ es el mismo orden de las clases 200, 209 y 217
```

Y la medida de si el centro funciona:

```
incidentes al día, y proporción que resulta ser algo
tiempo hasta la primera acción
y —la que más dice— PROPORCIÓN DE TÉCNICAS SIMULADAS QUE SE
DETECTAN
  → esa cifra es la única que mide la detección de verdad
```

4. Respuesta y coste

La respuesta, con la decisión de qué se automatiza:

```
SE AUTOMATIZA
  enriquecer el incidente: quién es esa identidad, qué
    recursos toca, de dónde viene esa dirección
  aislar un dispositivo comprometido
  revocar sesiones de una identidad
  bloquear una dirección
```

```
  → acciones reversibles y de bajo daño si el aviso es
    falso

NO SE AUTOMATIZA
  apagar producción
  borrar recursos
  bloquear cuentas de administración
  → un falso positivo con esas acciones es un incidente
    peor que el que se intentaba evitar

Y EN CUALQUIER CASO
  toda acción automática deja registro y es reversible
  y hay una forma de desactivar la automatización
    rápidamente                                clase 259
```

Y el procedimiento, con lo que este programa exige:

```
cada regla con procedimiento enlazado          clase 125
probado por alguien que no lo escribió           ley 22
y con el paso de contención al principio, no al final
                                                clase 216
```

El coste, que en seguridad se dispara con facilidad:

```
LAS DOS PARTIDAS
  planes de protección por recurso: por servidor, por
    base, por almacén
  ingesta y retención de las señales

Y LAS DECISIONES
  activar la protección donde hay datos o exposición, no
    en todo
  ingerir para detectar solo lo que alimenta reglas
  el resto, en plan barato para investigar
  y compromiso de capacidad si el volumen es estable

→ y hay que revisarlo cada trimestre, como el resto
                                                clase 214
```

Y una comprobación honesta que conviene hacer:

```
¿cuántos incidentes reales ha detectado este centro en el
último año?
¿y cuántos se detectaron por otra vía —un usuario, una
factura, un socio—?
→ la segunda cifra dice qué no cubre la detección
→ en la clase 200, la exfiltración de 14 meses la
  encontró un inventario, no una alerta
```

Y la lista de comprobación de la clase:

- ■ las recomendaciones se ordenan por alcance, no por puntuación
- ■ se usan los caminos de ataque para priorizar
- ■ lo arreglado se impide por política para que no vuelva
- ■ toda exención tiene motivo, dueño y caducidad
- ■ el marco de cumplimiento elegido es el que aplica
- ■ cada control tiene dueño y evidencia
- ■ lo manual está documentado aparte, sin inflar el porcentaje
- ■ los conectores se eligieron por valor y volumen
- ■ lo que alimenta reglas está en plan analítico
- ■ las reglas se calibraron en modo observación antes
- ■ cada regla tiene procedimiento enlazado
- ■ se simulan técnicas periódicamente y se mide qué proporción se detecta
- ■ las acciones automáticas son reversibles y registradas
- ■ hay forma rápida de desactivar la automatización
- ■ se mide cuántos incidentes se detectaron por otra vía
- ■ el coste de protección e ingesta se revisa cada trimestre

Y el cierre que enlaza con la clase siguiente: con seguridad, observabilidad y datos en pie, quedan el coste y la resiliencia, que en esta nube tienen herramientas propias y una para provocar fallos de forma controlada. Es la materia de la clase 227.

Ejemplo trabajado

CloudShop monta la seguridad operativa de Azure. Lo que sigue son las recomendaciones ordenadas por alcance en vez de por puntos, la simulación que reveló que la mitad de las técnicas no se detectaban, y la factura que hubo que recortar.

La postura, al empezar:

```
recomendaciones abiertas                          1.847
puntuación de seguridad                             47 %

ordenadas por PUNTOS, las cinco primeras
  1  cifrado en discos de máquinas               +7,2 pts
     412 recursos, casi todos de desarrollo
  2  copias de seguridad habilitadas             +6,1 pts
  3  versión de TLS mínima                       +5,4 pts
  4  agente de supervisión instalado             +4,8 pts
  5  cifrado en tránsito en almacenes            +4,1 pts

ordenadas por ALCANCE, las cinco primeras
  1  una máquina expuesta a internet con puerto de
     administración abierto, y desde su identidad se
     alcanza la base de pedidos
  2  3 identidades con Propietario en producción
                                                 clase 218
  3  2 cuentas de almacenamiento públicas con datos de
     clientes
  4  una base de datos con acceso público y contraseña de
     administrador antigua
  5  una identidad de canalización que puede asignar
     papeles

→ ninguna de las cinco de alcance estaba entre las 40
  primeras por puntos
```

Y el camino de ataque que la herramienta calculó:

```
internet
  → máquina de salto con puerto de administración abierto
    → su identidad administrada tiene Colaborador en la
      suscripción
      → alcanza la base de pedidos y el almacén de copias
        → 2,3 M de registros de clientes

pasos                                                 4
recursos comprometibles                          11.400
tiempo estimado de explotación                  minutos

→ ese único camino concentraba más riesgo que las 1.800
  recomendaciones restantes juntas
```

El orden de trabajo que se siguió:

```
semana 1   los 5 caminos de ataque: cerrados
           puerto de administración cerrado; acceso por
           servicio de administración temporal  clase 256
           identidades reducidas de ámbito      clase 218
           almacenes públicos cerrados
semana 2   políticas de denegación para que no vuelvan
                                                clase 217
semanas 3-9 remediación por lotes del resto
           tareas automáticas para lo que se puede
semana 10  exenciones escritas para lo que no compensa
           41 exenciones, todas con motivo, dueño y fecha

puntuación de seguridad          47 % → 81 %
recomendaciones abiertas       1.847 → 214
caminos de ataque con datos al final    5 → 0
```

Y la observación:

```
la puntuación subió 34 puntos
y los 34 puntos NO son la medida de lo que mejoró
→ lo que mejoró fue cerrar 5 caminos, que valían 0,9 puntos
```

```
  entre los cinco                                  ley 17
```

El cumplimiento, medido con honestidad:

```
marco elegido   el que exige el contrato con el mayor
                cliente empresarial; no se activaron los
                otros cuatro que ofrecía la herramienta

resultado presentado a auditoría
  controles del marco                              114
    automatizados y conformes                       71
    automatizados con excepciones documentadas      22
    manuales, con evidencia adjunta                 18
    NO cumplidos                                     3
      · rotación de claves de cifrado propias
      · segregación de funciones en dos procesos
      · registro de acceso físico (fuera de alcance:
        es del proveedor)

  y un plan con fecha para los 3

→ lo que la herramienta mostraba como «94 % conforme»
  ocultaba que 18 controles eran manuales y que 3 no se
  cumplían
```

La detección: la simulación que dolió.

```
se montaron 61 reglas de detección
la mayoría, plantillas del proveedor

la simulación, con 14 técnicas conocidas
  técnica                                  ¿detectada?
  fuerza bruta contra identidad                 sí
  inicio de sesión desde país inhabitual        sí
  creación de identidad con permisos altos      sí
  exfiltración a almacén externo                NO  ←
  desactivación de registro de auditoría        sí
  acceso a secretos desde identidad inusual     NO  ←
  enumeración de recursos                       NO  ←
  uso de credencial de emergencia               sí
  movimiento lateral por identidad administrada NO  ←
  borrado masivo de recursos                    sí
  cambio de política de seguridad               sí
  consultas de nombres anómalas                 NO  ←
  ejecución en contenedor con privilegios       NO  ←
  persistencia por credencial federada nueva    NO  ←

  detectadas                                 7 de 14
```

Y el análisis de las siete que no:

```
3  la regla existía y no se disparaba
   · umbral demasiado alto (enumeración: exigía 500
     llamadas en 1 min; el ataque hizo 80 en 10 min)
   · consulta con un error de campo
   · regla en estado deshabilitado desde su creación

3  no había regla
   · movimiento lateral por identidad administrada
   · persistencia por credencial federada nueva
   · ejecución en contenedor con privilegios

1  la señal NO SE INGERÍA
   · consultas de nombres: el conector estaba desactivado
     por coste                                clase 200
```

Y las correcciones:

```
umbrales ajustados con datos reales
reglas escritas para las 3 técnicas sin cobertura
conector de nombres activado, con plan básico para el
  volumen y regla sobre un resumen agregado
  → coste 340 €/mes en vez de 2.100 €
y simulación trimestral, con la cifra publicada

segunda simulación, 3 meses después     13 de 14
```

```
  la que falta: exfiltración a almacén externo por un canal
  que se decidió no cubrir; aceptada por escrito, con la
  mitigación de que el perímetro de datos lo impide
                                              clase 200
```

El ruido, calibrado:

```
al activar las 61 reglas de golpe
  incidentes el primer día                          890
  reales                                              2

calibración
  modo observación durante 3 semanas
  exclusiones legítimas
    · el escáner de vulnerabilidades propio
    · la cuenta de copias, que accede a todo por diseño
    · los rangos de las 11 oficinas
    · una integración de un socio
  reglas retiradas por no accionables                14

incidentes al día                               890 → 6
  que resultan ser algo                               1,4
  tiempo hasta la primera acción                8 min
```

El coste, recortado:

```
planes de protección por recurso          4.100 €     1.900 €
  → activados en todo; se dejaron donde hay datos o
    exposición
ingesta de señales de seguridad           5.800 €     1.240 €
  → registros de red: de analítico a básico, con
    resúmenes agregados para las reglas
  → identidad y actividad: analítico, sin cambios
retención                                 1.100 €       380 €
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
total                                    11.000 €     3.520 €
```

La comprobación honesta, al año:

```
incidentes reales detectados por el centro             9
incidentes detectados por OTRA vía                     4
  · una factura anómala (una prueba de carga olvidada)
  · un socio que avisó de un correo sospechoso
  · un inventario de destinos de salida       clase 200
  · un usuario que reportó un comportamiento raro

→ 4 de 13: casi un tercio no lo vio la detección
→ y las cuatro vías son las de siempre: coste, terceros,
  inventarios y personas
```

La lección que esta clase deja: ordenar por puntuación habría hecho al equipo cifrar cuatrocientos doce discos de desarrollo mientras un camino de cuatro pasos llevaba de internet a dos millones trescientos mil registros de clientes. Y la simulación reveló que siete de catorce técnicas no se detectaban, aunque las reglas existían: tres tenían umbrales o errores, tres no existían y una dependía de una señal que se había desactivado por coste.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-18-azure-production-architecture/226-defender-for-cloud-policy-y-sentinel/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 azure-security-operations en evidence/ usando la plantilla indicada.
5. Para proveedor real, sigue sandbox.requires, registra costo y ejecuta destroy.

Evidencia esperada

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

Reto verificable

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

■ Criterio de aceptación

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

■■ Errores frecuentes

Síntoma	Causa probable	Corrección
El equipo trabaja mucho en seguridad y el riesgo real no baja	Las recomendaciones se priorizan por la puntuación agregada	Ordena por alcance desde el punto comprometido y usa los caminos de ataque calculados.
Una recomendación resuelta reaparece meses después	Se corrigió el recurso pero nada impide crearlo mal otra vez	Tras remediar, impide por política que vuelva a ocurrir.
El informe dice que se cumple el 94 % y la auditoría encuentra huecos	El porcentaje incluye controles manuales sin evidencia y oculta los no cumplidos	Presenta el desglose por tipo de control, con dueño y evidencia, y un plan con fecha para los no cumplidos.
Las reglas de detección existen y el ataque no se detecta	Umbrales mal puestos, consultas con errores o reglas deshabilitadas	Simula técnicas conocidas periódicamente y publica qué proporción se detecta; es la única medida real.
El centro genera cientos de incidentes al día y nadie los mira	Las reglas se activaron sin calibrar ni excluir lo legítimo	Despliega en modo observación, excluye lo conocido, retira lo no accionable y mide la proporción que resulta ser algo.
La factura de seguridad es enorme	Protección activada en todos los recursos e ingesta de señales voluminosas en plan analítico	Protege donde hay datos o exposición, ingiere en plan analítico solo lo que alimenta reglas y usa resúmenes agregados para lo voluminoso.

■ 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 cómo se relacionan?
2. ¿Por qué la puntuación de seguridad es mala guía de prioridad?
3. ¿Qué debe acompañar a cada control de cumplimiento para que la medida sea honesta?
4. ¿Qué mide de verdad si una detección funciona?

5. ¿Qué acciones de respuesta conviene automatizar y cuáles no?

Referencias

- Microsoft (2025). Microsoft Defender for Cloud: secure score and attack paths. <https://learn.microsoft.com/en-us/azure/defender-for-cloud/secure-score-security-controls>
- Microsoft (2025). Regulatory compliance in Defender for Cloud. <https://learn.microsoft.com/en-us/azure/defender-for-cloud/regulatory-compliance-dashboard>
- Microsoft (2025). Microsoft Sentinel: data connectors and analytics rules. <https://learn.microsoft.com/en-us/azure/sentinel/overview>
- MITRE (2025). ATT&CK for cloud. <https://attack.mitre.org/matrices/enterprise/cloud/>
- Microsoft (2025). Sentinel cost optimization and data tiers. <https://learn.microsoft.com/en-us/azure/sentinel/billing>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 18 en PDF · Recorrido de Azure en PDF · Manual integral

Anterior	Índice	Siguiente
← 225 · Azure Monitor, Application Insights y OpenTelemetry	Parte 18 · Programa	227 · Cost Management, Advisor, resiliencia y Chaos Studio →

Evaluación — 226 Defender for Cloud, Policy y Sentinel

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 azure-security-operations con diagrama o configuración, evidencia de ejecución, alternativa descartada, riesgo residual y criterio de rollback.

Criterio de aceptación

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

Escala

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

227 — Cost Management, Advisor, resiliencia y Chaos Studio

Parte: 18 — Azure: arquitectura empresarial y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: finops · Estado: EXECUTABLECORE

Propósito

Cerrar la operación de Azure con las dos cosas que se dejan para el final y deciden si el sistema es sostenible: el control del coste y la comprobación de la resiliencia. La clase aplica el método de la clase 214 con las herramientas de esta nube, evalúa las recomendaciones automáticas con escepticismo —el asesor propone reservas sobre cargas que había que retirar—, y usa la inyección controlada de fallos para comprobar lo que la clase 215 exige comprobar.

Resultados de aprendizaje

Al finalizar podrás:

1. Atribuir el coste con etiquetas impuestas en la creación.
2. Evaluar las recomendaciones automáticas antes de aplicarlas.
3. Comprometer capacidad solo sobre la base estable y vigilar su uso.
4. Inyectar fallos de forma controlada para comprobar la resiliencia.
5. Cerrar el ciclo: cada hallazgo, una acción con dueño y fecha.

Conceptos centrales

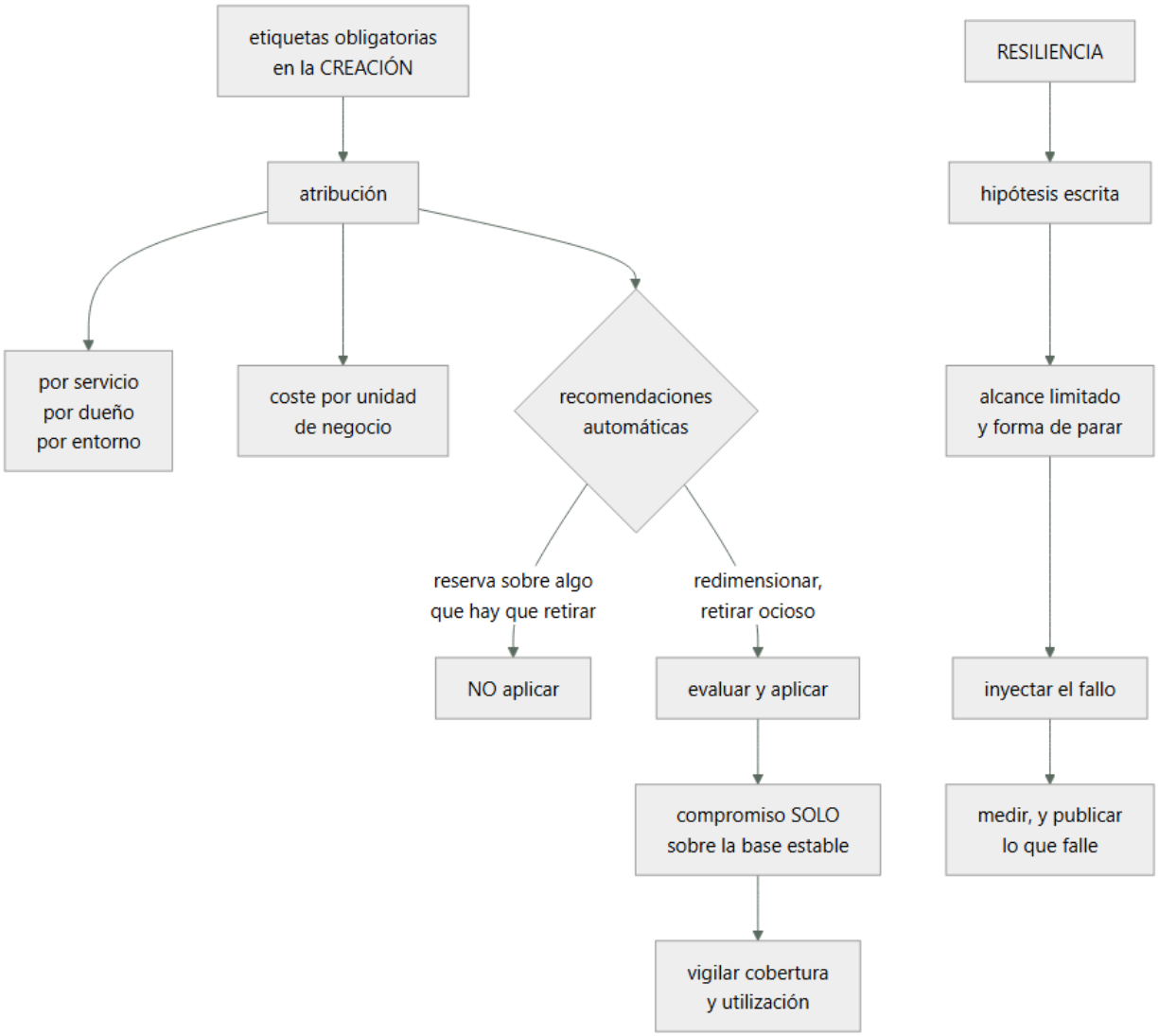
Concepto	Comprensión verificable
ámbito de facturación	Nivel al que se agrega el gasto: cuenta de facturación, perfil, sección y suscripción.
etiqueta heredada	Etiqueta que se propaga del grupo de recursos al recurso. No todos los servicios la respetan.
reserva de capacidad	Compromiso de uso a uno o tres años a cambio de descuento. Se aplica a la base estable.
plan de ahorro	Compromiso de gasto por hora, más flexible que la reserva y con menos descuento.
experimento de caos	Inyección controlada de un fallo, con hipótesis, alcance limitado y forma de parar.
recomendación automática	Sugerencia de la plataforma sobre coste, fiabilidad o rendimiento. Se evalúa, no se aplica.

Modelo mental

La arquitectura Azure nace del tenant y las políticas heredables, y llega al workload mediante identidades administradas y redes privadas.

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. Atribuir el coste en esta nube

El método es el de la clase 214 y aquí hay particularidades que hay que conocer.

```
LA JERARQUÍA DE FACTURACIÓN
  cuenta de facturación → perfil → sección → suscripción
  → el gasto se agrega por ahí, y por eso la separación
    por suscripción de la clase 217 es también una
    decisión de facturación

LAS ETIQUETAS
  obligatorias en la creación, con valores cerrados, por
  política                                    clase 217
  → y la herencia del grupo de recursos NO la respetan
    todos los servicios
  → hay que comprobar cuáles, y para esos, etiquetar el
    recurso directamente
```

Y el gasto que no se puede etiquetar, que aquí es apreciable:

```
transferencia entre zonas y regiones
servicios compartidos: centro de red, cortafuegos,
  resolutor, área de observabilidad
y los recursos gestionados por la plataforma

→ se reparten por regla escrita y acordada
→ y hay que decir cuánto se reparte y con qué criterio
```

```
                                                  clase 214
```

Las cinco vistas de la clase 214, con lo que suele aparecer en Azure:

```
POR SERVICIO      el 80 % en 3 o 4 líneas
POR DUEÑO         y cuánto no lo tiene
POR ENTORNO       preproducción por encima del 25 % de
                  producción es señal
POR TIPO DE USO   dentro de cada servicio grande
POR UNIDAD DE
  NEGOCIO         la única cifra comparable

y las partidas que sorprenden en esta nube
  unidades de petición de la base distribuida  clase 223
  ingesta de observabilidad y de seguridad clases 225, 226
  puntos privados e inspección del cortafuegos clase 219
  transferencia entre zonas
  y los planes de servicio con instancias encendidas y sin
    tráfico                                    clase 221
```

Y los presupuestos con acción, igual que antes:

```
por suscripción y por servicio, sobre previsión
en entornos no productivos, con acción real
en producción, solo aviso
y detección de anomalías por servicio y por dueño
                                                  clase 214
```

2. Las recomendaciones automáticas, con escepticismo

La plataforma propone mejoras de coste, fiabilidad, rendimiento y seguridad. Son útiles y no se aplican sin evaluar.

```
LAS DE COSTE, y su trampa
  «compra una reserva para estas máquinas»
  → calcula el ahorro suponiendo que ESAS máquinas seguirán
    existiendo tres años
  → y no sabe que estaban en la lista de retirada
  → comprometerse con lo que hay que apagar es la peor
    compra posible                              ley 25

  «redimensiona esta máquina infrautilizada»
  → mira el uso medio, no el pico ni la estacionalidad
  → hay que comprobar el percentil alto antes  clase 212

  «retira este recurso ocioso»
  → suele ser correcta, y aun así: apagar antes de borrar
                                                  clase 214
```

Y el orden correcto, que evita comprar lo que hay que tirar:

```
1  RETIRAR lo que no se usa
2  REDIMENSIONAR lo sobredimensionado
3  APAGAR por horario lo no productivo
4  Y SOLO ENTONCES comprometer capacidad

→ comprometer primero congela el desperdicio durante años
```

Los compromisos, con la decisión entre los dos tipos:

```
RESERVA DE CAPACIDAD
  se compromete un tipo y tamaño concreto
  + descuento mayor
  − rígida: si se cambia de familia, el descuento se pierde
    o hay que intercambiarla

PLAN DE AHORRO
  se compromete un gasto por hora
  + flexible: se aplica a lo que haya
  − descuento menor

CRITERIO
  infraestructura estable y previsible   → reserva
  cargas que van a cambiar de forma      → plan de ahorro
  y en ambos casos, SOLO sobre la base estable medida con
  meses de histórico                          clase 143
```

Y lo que hay que vigilar después:

```
COBERTURA     qué proporción del uso está cubierta
UTILIZACIÓN   qué proporción de lo comprometido se usa
  → por debajo del 95 %, es dinero tirado
  → y baja sola cuando se apaga algo o se cambia de
    familia

→ alerta si la utilización cae por debajo del umbral
→ y revisión antes de cada renovación
```

Y las recomendaciones de fiabilidad, que merecen más atención que las de coste:

```
«esta base no tiene redundancia de zona»
«este recurso no tiene copias de seguridad»
«esta configuración no sobrevive a la pérdida de una zona»

→ estas se leen con la aritmética de la clase 185: ¿cambian
  el techo de disponibilidad del flujo crítico?
→ y si sí, se aplican; si no, se aceptan por escrito
```

3. Inyectar fallos de forma controlada

La clase 215 exige comprobar la resiliencia. La inyección controlada de fallos es la forma de hacerlo sin esperar a un incidente.

```
QUÉ PERMITE INYECTAR
  apagar o reiniciar máquinas y nodos
  presión de CPU, memoria o disco
  latencia y pérdida en la red
  bloqueo del acceso a un servicio dependiente
  conmutación forzada de una base
  caída de una zona completa
  y fallos de una dependencia de la plataforma
```

Y la disciplina, que es la de la clase 215:

```
HIPÓTESIS ESCRITA, antes
  «al perder una zona, el flujo de compra seguirá
   funcionando con latencia menor de 600 ms y sin errores;
   la búsqueda quedará degradada»
  → lo valioso es dónde falla la hipótesis

ALCANCE LIMITADO
  un porcentaje de instancias, una zona, un servicio
  → y el experimento debe poder pararse en segundos

VENTANA Y OBSERVACIÓN
  con negocio avisado y con alguien tomando notas
  → y con tráfico real, no de madrugada la primera vez

ESCALA
  1  en papel: recorrer el procedimiento
  2  en preproducción
  3  en producción con alcance pequeño
  4  en producción completo
```

Los experimentos que más encuentran, por orden:

```
1  PÉRDIDA DE UNA ZONA
   → comprueba que las réplicas están repartidas de verdad
   → y que la capacidad restante aguanta   clases 185, 212

2  DEPENDENCIA QUE RESPONDE LENTO, no que cae
   → el fallo gris; la mayoría de los diseños no lo
     resisten                                  clase 185
   → y es donde se descubre que una dependencia declarada
     blanda es dura                            clase 201

3  CONMUTACIÓN DE BASE DE DATOS
   → cronometra y mide la pérdida real       clase 223

4  PRESIÓN DE RECURSOS
   → descubre límites de memoria mal puestos  clase 213

5  PÉRDIDA DE UN SERVICIO DE PLATAFORMA
```

```
   → el almacén de secretos, el resolutor, el registro de
     imágenes
   → y estos son los que nadie prueba y los que paran todo
                                                clase 215
```

Y la disciplina posterior, que es lo que hace útil el ejercicio:

```
cada hallazgo → una acción con dueño y fecha
el procedimiento se corrige EN EL MOMENTO
y el experimento se repite hasta que pase entero

→ un experimento que encuentra problemas y no se repite ha
  servido para la mitad                        clase 215
```

Y una advertencia sobre el permiso y el alcance:

```
la herramienta necesita permisos para romper cosas
→ y esos permisos son peligrosos si quedan permanentes
→ elevación temporal para ejecutar experimentos
                                                clase 218
→ y el alcance del experimento, restringido por etiqueta o
  por grupo de recursos
```

4. Cerrar el ciclo

Coste y resiliencia comparten una propiedad: son las dos cosas que se degradan solas si nadie las mira.

```
EL COSTE SE DEGRADA PORQUE
  se crean recursos y no se retiran               ley 25
  se sobredimensiona por si acaso
  se acumulan entornos y copias
  y nadie tiene el gasto entre sus señales

LA RESILIENCIA SE DEGRADA PORQUE
  el sistema crece y las comprobaciones no      clase 216
  se añaden dependencias sin recalcular el techo clase 185
  y los procedimientos envejecen                  ley 22
```

Y el ritmo que este programa propone:

```
SEMANAL
  revisión de anomalías de coste abiertas
  recursos ociosos detectados y retirados

MENSUAL
  coste por unidad de negocio y su tendencia
  cobertura y utilización de compromisos
  cumplimiento de las iniciativas             clase 217

TRIMESTRAL
  un experimento de resiliencia, ejecutado
  simulación de técnicas de ataque            clase 226
  prueba del acceso de emergencia             clase 218
  revisión de accesos
  y revisión de exenciones vencidas

ANUAL
  ejercicio de pérdida de región completo     clase 215
  y revisión de las decisiones registradas    clase 190
```

Y las señales que dicen si el sistema está sano:

```
COSTE
  proporción de gasto atribuido           > 90 %
  coste por unidad de negocio             tendencia
  utilización de compromisos              > 95 %
  recursos ociosos vivos                  tendencia a cero

RESILIENCIA
  proporción de pruebas negativas que pasan
  plazo de recuperación MEDIDO, no declarado
  proporción de técnicas simuladas detectadas clase 226
  y experimentos ejecutados frente a planificados
```

Y la lista de comprobación de la clase:

- ■ las etiquetas obligatorias se imponen en la creación

- se comprobó qué servicios no heredan etiquetas
- el coste no etiquetable se reparte por regla escrita
- hay presupuestos con acción en entornos no productivos
- hay detección de anomalías por servicio y por dueño
- se retiró y redimensionó ANTES de comprometer capacidad
- los compromisos cubren solo la base estable medida
- se vigilan cobertura y utilización, con alerta
- las recomendaciones automáticas se evalúan, no se aplican
- las de fiabilidad se leen con la aritmética del techo
- hay experimentos de caos con hipótesis y forma de parar
- el alcance de los experimentos está restringido
- los permisos para inyectar fallos son temporales
- cada hallazgo tiene dueño y fecha, y el experimento se repite
- existe un calendario semanal, mensual, trimestral y anual

Y el cierre que enlaza con la clase siguiente: con todo lo de esta parte montado, queda ponerlo junto en un sistema productivo y comprobarlo. Es la materia de la clase 228, que además cierra la parte 18.

Ejemplo trabajado

CloudShop cierra la operación de su plataforma en Azure. Lo que sigue es la reserva que casi se compra sobre máquinas que había que retirar, y el experimento de pérdida de zona que encontró seis problemas, ninguno de infraestructura.

El coste, al revisar:

```
total mensual                              31.400 €

por servicio
  base distribuida                          5.740   18 %
  cómputo (planes y contenedores)           4.900   16 %
  observabilidad y seguridad                4.780   15 %
  transferencia y red                       4.210   13 %
  bases relacionales                        3.100   10 %
  almacenamiento                            2.840    9 %
  resto                                     5.830   19 %

por dueño
  atribuido                                19.100   61 %
  sin atribuir                             12.300   39 %
```

Y el detalle de lo no atribuido:

```
transferencia y servicios compartidos          6.100
recursos sin etiqueta                          3.240
  → de ellos, 1.870 € en servicios que NO HEREDAN las
    etiquetas del grupo de recursos
  → el equipo creía que las heredaban todos
recursos ociosos                               2.960
```

Y el hallazgo de la herencia:

```
se comprobó servicio por servicio cuáles heredan
  heredan                                        61 %
  NO heredan                                     39 %
    · cuentas de almacenamiento en algunas operaciones
    · discos creados por conjuntos de escalado
    · direcciones públicas creadas automáticamente
    · instantáneas y copias

corrección
  política de tipo MODIFICAR que añade las etiquetas al
  crear, en vez de confiar en la herencia      clase 217
  → 3.240 € pasaron a estar atribuidos en 3 semanas
```

La reserva que no se compró.

```
la recomendación automática decía
  «reserva a 3 años para 22 máquinas de la familia D»
  ahorro estimado                         4.100 €/mes
  coste del compromiso                   38 % del total

lo que el equipo comprobó antes de comprar
```

```
  ¿qué son esas 22 máquinas?
    14  nodos del clúster antiguo, en la lista de retirada
        del trimestre siguiente
     5  máquinas del sistema heredado de facturación, con
        migración planificada a 18 meses          clase 223
     3  máquinas de un proyecto terminado en 2024, que
        nadie había apagado                        ley 25

→ 17 de 22 no existirían en un año
→ comprar la reserva habría congelado el desperdicio
  durante tres

lo que se hizo, en el orden correcto
  1  RETIRAR      las 3 del proyecto terminado
                  y 6 más detectadas al inventariar
                                        -1.240 €/mes
  2  REDIMENSIONAR  11 máquinas con uso p95 muy por debajo
                    de su tamaño         -680 €/mes
  3  APAGAR POR HORARIO  no producción
                                        -2.100 €/mes
  4  Y ENTONCES comprometer
     plan de ahorro (no reserva) sobre la base estable
     medida con 4 meses de histórico
     motivo del plan y no la reserva: el clúster antiguo se
     retira y la familia cambiará       clase 143
                                        -1.900 €/mes

ahorro total                            -5.920 €/mes
frente a los 4.100 € de la recomendación, y sin
comprometerse con lo que había que tirar
```

El experimento de pérdida de zona.

```
HIPÓTESIS, escrita
  1  el flujo de compra seguirá funcionando
  2  la latencia p99 subirá por debajo de 600 ms
  3  no habrá errores para el usuario
  4  la búsqueda quedará degradada, no caída
  5  la recuperación al restaurar la zona será automática

ALCANCE   la zona 3 de las 3, en producción, con el 30 %
          del tráfico de un día normal
PARADA    detener el experimento restaura en < 60 s
OBSERVACIÓN  5 personas, 1 tomando notas

EJECUCIÓN, martes 15:00
```

Y lo que ocurrió:

```
15:00:00  se corta la zona 3

15:00:12  el balanceador retira los destinos de esa zona
          ✓ predicción 1: el flujo sigue

15:00:40  latencia p99                      410 ms
          ✓ predicción 2

15:01:10  aparecen errores 503 en el 3 % de las peticiones
          ✗ predicción 3
          causa   el plan de servicio tenía 6 instancias,
                  2 por zona; al perder 2, las 4 restantes
                  quedaron al 91 % de su codo  clase 186
                  y el escalado tardó 4 minutos
          → faltaba margen para perder una zona de tres
                                              clase 212

15:02:30  la BÚSQUEDA cayó por completo
          ✗ predicción 4
          causa   sus 3 réplicas estaban las tres en la
                  zona 3
                  el reparto entre zonas no estaba
                  configurado                clase 213
          → y nadie lo sabía porque nunca se había perdido
            una zona
```

```
15:04:00  la base distribuida siguió sirviendo
          ✓  (redundancia de zona activada)

15:04:20  el almacén de SECRETOS dejó de responder para 2
          servicios
          ✗ no estaba en la hipótesis
          causa   esos 2 servicios cacheaban el secreto al
                  arrancar; las instancias nuevas no
                  pudieron obtenerlo porque el punto
                  privado del almacén estaba en la zona 3
                                              clase 219

15:09:00  el equipo decide parar
15:09:40  zona restaurada

15:10-15:24  la recuperación NO fue automática
          ✗ predicción 5
          2 servicios quedaron con instancias marcadas como
          no sanas y hubo que reiniciarlas a mano
```

Los seis hallazgos:

#	hallazgo	tipo	acción
1	sin margen para perder una zona	capacidad	6 → 9 instancias
2	las 3 réplicas del buscador en la misma zona	reparto	restricción de reparto
3	punto privado del almacén de secretos en una sola zona	red	3 puntos, uno por zona
4	2 servicios cachean el secreto y no lo renuevan	código	renovación periódica
5	recuperación no automática	proceso	comprobación de salud corregida
6	nadie sabía qué hacer al ver los 503: no había procedimiento	proceso	escrito y probado

de infraestructura	0
de configuración, reparto, código y proceso	6

Y la segunda ejecución, dos meses después:

latencia p99 máxima	380 ms	✓
errores para el usuario	0	✓
búsqueda	degradada	✓
almacén de secretos	disponible	✓
recuperación	automática	✓
duración total del impacto	0 min	

El calendario que quedó:

semanal	anomalías de coste; ociosos retirados	
mensual	coste por pedido; cobertura y utilización; cumplimiento de iniciativas	
trimestral	1 experimento de resiliencia	
	simulación de técnicas de ataque	clase 226
	prueba de acceso de emergencia	clase 218
	revisión de accesos y de exenciones	
anual	ejercicio de pérdida de región	clase 215
	revisión de decisiones registradas	clase 190

El resultado, seis meses después:

coste mensual	31.400 €	23.100 €
coste atribuido	61 %	94 %
coste por pedido	0,071 €	0,049 €
utilización del compromiso	—	98 %
recursos ociosos	2.960 €	0 €
réplicas repartidas entre zonas	parcial	total
experimentos ejecutados	0	3
hallazgos con dueño y fecha	—	14

```
cerrados                                  —          13
```

La lección que esta clase deja: la recomendación automática proponía comprometerse tres años con veintidós máquinas de las que diecisiete no iban a existir en uno, y hacer las cosas en el orden correcto —retirar, redimensionar, apagar, y solo entonces comprometer— dio un ahorro mayor sin atarse a nada. Y el experimento de pérdida de zona encontró seis problemas, ninguno de infraestructura: los tres más graves fueron réplicas mal repartidas, un punto privado en una sola zona y dos servicios que cacheaban un secreto y no lo renovaban.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-18-azure-production-architecture/227-cost-management-advisor-resiliencia-y-chaos-studio/lab.py
```

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

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

Evidencia esperada

El artefacto principal es un cálculo trazable con unidad, supuesto y sensibilidad. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye azure-optimization 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
Parte del gasto sigue sin atribuir pese a tener etiquetas obligatorias	Se confía en la herencia del grupo de recursos y hay servicios que no la respetan	Comprueba qué servicios heredan y usa una política de tipo modificar que etiquete al crear.
Se compra un compromiso y la utilización cae a los pocos meses	Se comprometió capacidad sobre cargas que iban a retirarse o a cambiar de forma	Retira, redimensiona y apaga primero; compromete solo la base estable medida y usa plan de ahorro si la forma va a cambiar.
Aplicar una recomendación de redimensionado degrada el servicio	La recomendación usa el uso medio y no el pico ni la estacionalidad	Comprueba el percentil alto y el comportamiento en campaña antes de aplicar.
Al perder una zona el servicio se degrada más de lo previsto	No hay margen de capacidad para perder una zona de tres	Dimensiona para que la capacidad restante quede por debajo del codo, y comprueba con un experimento.
Todas las réplicas de un servicio están en la misma zona	No hay restricción de reparto y el planificador las agrupó	Declara restricciones de reparto por zona y compruébalo inyectando la pérdida de una.
Un experimento de caos se convierte en un incidente	Alcance amplio, sin hipótesis ni forma de parar, y permisos permanentes para romper cosas	Hipótesis escrita, alcance restringido por etiqueta, parada en segundos y permisos por elevación temporal.

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 no todas las etiquetas se heredan y qué hacer al respecto?
2. ¿En qué orden hay que actuar antes de comprometer capacidad?
3. ¿Cuándo conviene una reserva y cuándo un plan de ahorro?
4. ¿Qué cinco experimentos de resiliencia encuentran más problemas?
5. ¿Qué ritmo de revisión propone esta clase y qué se hace en cada periodo?

Referencias

- Microsoft (2025). Azure Cost Management and Billing. <https://learn.microsoft.com/en-us/azure/cost-management-billing/>
- Microsoft (2025). Azure Advisor recommendations. <https://learn.microsoft.com/en-us/azure/advisor/advisor-overview>
- Microsoft (2025). Azure savings plans and reservations. <https://learn.microsoft.com/en-us/azure/cost-management-billing/savings-plan/savings-plan-compute-overview>
- Microsoft (2025). Azure Chaos Studio. <https://learn.microsoft.com/en-us/azure/chaos-studio/chaos-studio-overview>
- Microsoft (2025). Reliability and availability zones in Azure. <https://learn.microsoft.com/en-us/azure/reliability/availability-zones-overview>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 18 en PDF · Recorrido de Azure en PDF · Manual integral

Evaluación — 227 Cost Management, Advisor, resiliencia y Chaos Studio

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 azure-optimization 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.

228 — Proyecto: CloudShop productivo en Azure

Parte: 18 — Azure: arquitectura empresarial y operación en producción Nivel: avanzado · Horas estimadas: 8
Laboratorio: capstone · Estado: EXECUTABLECORE

Propósito

Poner en producción el sistema completo de CloudShop en Azure con lo de las once clases anteriores, y comprobarlo con las pruebas negativas de toda la parte. La clase da el orden, el entregable y los criterios. Y cierra la parte 18: corrige las cinco predicciones de la clase 216 —tres acertadas y dos a medias—, actualiza el recuento de leyes, añade la ley 27 y escribe la hipótesis de la parte 19, contestando además la pregunta incómoda que la clase 216 se dejó planteada.

Resultados de aprendizaje

Al finalizar podrás:

1. Construir el sistema completo en el orden que evita rehacer.
2. Comprobar con las pruebas negativas de toda la parte.
3. Comparar el resultado con el equivalente en AWS, con cifras.
4. Corregir las cinco predicciones de la clase 216 con evidencia.
5. Escribir la hipótesis de la parte 19 en forma refutable.

Conceptos centrales

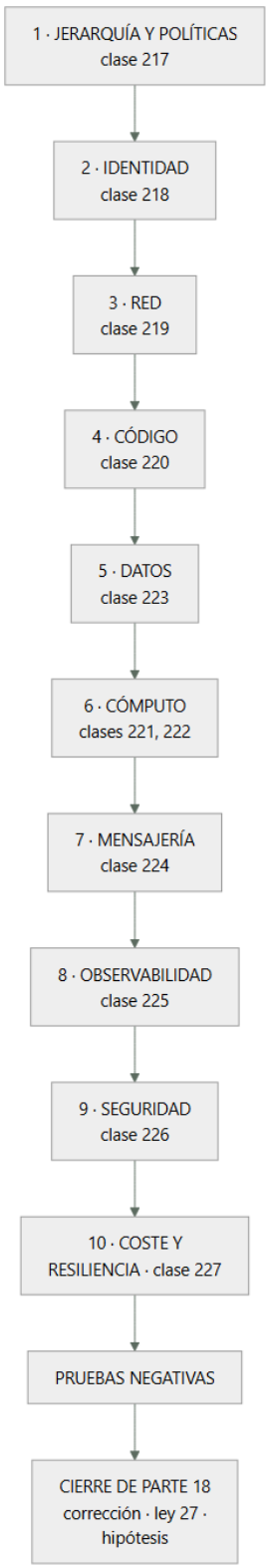
Concepto	Comprensión verificable
ley 27	Un control solo actúa sobre lo que cambia; lo que ya existe sigue incumpliendo hasta que alguien lo toque.
remediación	Tarea que corrige recursos existentes que incumplen. Es lo único que cierra la brecha que deja la ley 27.
equivalencia entre nubes	Correspondencia de conceptos. Alta en lo técnico, baja en el modelo operativo.
coste de traslado	Lo que cuesta llevar la misma arquitectura a otra nube. Se paga más en operación que en código.
prueba negativa de parte	Comprobación acumulada de las once clases, ejecutada sobre el sistema entero.
hipótesis de parte	Afirmación refutable escrita antes de estudiar, que la parte siguiente corrige con evidencia.

Modelo mental

La arquitectura Azure nace del tenant y las políticas heredables, y llega al workload mediante identidades administradas y redes privadas.

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

■ Flujo de razonamiento



Desarrollo

1. El encargo, el orden y el entregable

El encargo. Llevar a producción la plataforma de pedidos de CloudShop en Azure, con usuarios reales, guardia, presupuesto y continuidad comprobada.

El orden, por coste de cambio:

```
1  JERARQUÍA Y POLÍTICAS                          clase 217
   grupos de administración, suscripciones por carga y
   entorno, iniciativas en auditoría antes de denegar
   → lo que llega tarde deja recursos incumpliendo
                                                    ley 27

2  IDENTIDAD                                      clase 218
   identidades administradas y federadas, sin secretos
   asignaciones en el ámbito del recurso
   elevación temporal y acceso de emergencia probado

3  RED                                            clase 219
   centro y radios, salida centralizada
   puntos privados con zonas DNS enlazadas a TODAS las
   redes

4  CÓDIGO                                         clase 220
   separado por ciclo de vida y ámbito, con módulos
   verificados y pilas de despliegue

5  DATOS                                          clase 223
   familia por patrones, consistencia por operación,
   índices recortados

6  CÓMPUTO                              clases 221, 222
   la opción gestionada que corresponda; clúster solo donde
   hace falta

7  MENSAJERÍA                                     clase 224
   el servicio que garantiza lo que hace falta, con
   idempotencia y fallidos vigilados

8  OBSERVABILIDAD                                 clase 225
   diagnóstico por política, instrumentación estándar

9  SEGURIDAD                                      clase 226
   postura por alcance, detección comprobada con simulación

10 COSTE Y RESILIENCIA                            clase 227
   atribución, compromisos tras retirar, experimentos
```

El entregable, con las piezas propias de esta parte:

```
1  el problema, con su cifra
2  jerarquía y iniciativas, con el estado de cumplimiento
3  inventario de identidades y su ALCANCE medido
4  topología de red, con las zonas DNS y su enlace
5  el código, con lo que gestiona cada pila
6  patrones de acceso y niveles de consistencia por
   operación
7  la lista de VALORES POR DEFECTO cambiados
8  observabilidad: qué se ingiere, en qué plan y qué cuesta
9  detección: qué técnicas se simulan y cuáles se detectan
10 coste atribuido y por unidad de negocio
11 experimentos de resiliencia y sus hallazgos
12 pruebas negativas, con los fallos publicados
13 lo que NO se hace, y por qué
```

Las pruebas negativas de la parte:

- quitar una política heredada desde una suscripción
- crear un recurso sin etiquetas obligatorias
- crear un recurso en una región no permitida
- obtener la identidad de otro espacio de nombres
- asumir una identidad federada desde otro repositorio
- activar un papel privilegiado sin aprobación
- entrar con la cuenta de acceso de emergencia
- alcanzar un radio desde otro radio
- resolver un nombre con punto privado desde cada red
- alcanzar un servicio con punto privado desde internet
- sacar datos a un destino no declarado
- desplegar quitando un recurso de la plantilla
- borrar la pila y comprobar que los datos sobreviven
- enviar el mismo mensaje 50 veces

- dejar un mensaje en la cola de fallidos y esperar la alerta
- perder una zona completa
- simular 14 técnicas de ataque
- provocar la condición de cada alerta
- desplegar el entorno de cero en una suscripción vacía

Y los criterios de evaluación, con el peso donde importa:

1	las políticas están en grupos de administración	3
2	se midió el cumplimiento antes de denegar	2
3	ninguna asignación amplia sin justificar	3
4	no quedan secretos de cliente sin excepción	3
5	las zonas DNS están enlazadas a todas las redes	3
6	la consistencia se decide por operación	2
7	la lista de valores por defecto cambiados es explícita	3
8	la instrumentación es estándar	2
9	se simulan técnicas y se publica la proporción	3
10	el coste atribuido supera el 90 %	2
11	hay experimentos de resiliencia ejecutados	3
12	las pruebas negativas se ejecutaron y hay fallos publicados	3

2. Cierre de la parte 18: corrección de las cinco predicciones

Las cinco predicciones de la clase 216, corregidas con la evidencia de las clases 217 a 227.

1. «los valores por defecto de Azure estarán mal elegidos en proporción parecida a los de AWS, pero fallarán en otro sitio: los de AWS son permisivos en red y almacenamiento; los de Azure lo serán en identidad y ámbito de permisos»

   PRIMERA MITAD CORRECTA: la lista de valores por defecto que hubo que cambiar es comparable a la de AWS. SEGUNDA MITAD FALLADA. Los peores valores por defecto de Azure no fueron de identidad: fueron de COSTE y de OBSERVABILIDAD. La consistencia fuerte del asistente costaba 1.720 €/mes; la política que indexa todos los campos, otros 1.020; y el 93 % de los recursos no enviaba ningún registro porque la configuración de diagnóstico no existe si no se crea. El problema de identidad fue real y grave, pero no era un valor por defecto: era un error de configuración que alguien tomó por comodidad.

2. «la jerarquía será el equivalente del plan de direcciones: se decide en una tarde, condiciona una década, y renumerarla costará meses»

   A MEDIAS, y la diferencia importa. Se decidió mal —47 suscripciones colgando de la raíz— y condicionó todo, como el plan de direcciones. Pero el mecanismo del coste es OTRO: mover las 61 suscripciones al grupo correcto fue instantáneo y gratis. Lo que costó cuatro meses fue corregir los recursos que ya existían, porque las políticas de denegación no actúan hacia atrás. En redes, lo caro es renumerar; aquí, lo caro es que el control nuevo no toca lo viejo. Y eso es una ley distinta.

3. «la identidad será el eje; el error más frecuente será un ámbito de asignación demasiado amplio, el equivalente del comodín de la clase 206»

   CORRECTA, y de forma casi literal. De 1.940 asignaciones, 820 estaban en la suscripción y 31 en grupos de administración; una sola identidad de canalización alcanzaba 14.200 recursos. Y en el clúster, la credencial federada atada solo al emisor permitía a cualquier cuenta de servicio de cualquier espacio de nombres obtenerla. El mecanismo cambia de nombre en cada nube y produce el mismo resultado.

4. «la mayoría de los conceptos se corresponderán uno a uno; lo que no se corresponderá es el modelo operativo, y trasladar la misma arquitectura costará más en operación

que en código»

CORRECTA. Se correspondieron: federación de identidad, puntos privados, centro y radios, infraestructura como código, contenedores, mensajería, observabilidad y coste. NO se correspondieron: el modelo de políticas, que aquí es mucho más fuerte y no actúa hacia atrás; el modelo de ámbitos de asignación; los cinco niveles de consistencia, que no tienen equivalente; los modos de despliegue; y la necesidad de crear el diagnóstico recurso a recurso. El código se tradujo en semanas; entender esas cinco diferencias operativas costó los tres primeros meses.

5. «los problemas volverán a ser de las leyes 25, 15 y 22; y si acierta por cuarta vez consecutiva, dejará de tener mérito y habrá que preguntarse por qué, sabiéndolo, sigue ocurriendo»

CORRECTA por cuarta vez, y toca contestar la pregunta. La evidencia: cuatro políticas quitadas por equipos sin que nadie se enterara; zonas DNS enlazadas a 4 de 43 redes durante siete meses; una configuración sin marcar como específica de ranura; el 93 % de los recursos sin diagnóstico; siete de catorce técnicas sin detectar con las reglas creadas; tres máquinas de un proyecto terminado en 2024.

Y la respuesta a la pregunta, que es lo que esta clase debe a la anterior:

```
las leyes 25, 15 y 22 no describen ERRORES
describen el ESTADO POR DEFECTO de cualquier sistema que
cambia

  lo provisional se queda        porque retirar es trabajo
                                 y nadie lo pide
  la señal no se mira            porque mirar es trabajo
                                 continuo y no urgente
  el procedimiento no funciona   porque ejecutarlo es
                                 trabajo y no hay incidente

→ no hace falta que nadie se equivoque para que se cumplan
→ hace falta que nadie haga algo, y eso ocurre siempre

y por eso saberlas no basta
  el equipo de este programa las conocía en las cuatro
  partes y las cuatro veces volvieron a ocurrir
  → lo único que las contrarresta es que la comprobación
    sea AUTOMÁTICA y PERIÓDICA
  → una persona que conoce la ley 15 sigue sin mirar el
    panel
  → una alerta de antigüedad, no
```

Marcador: tres correctas, dos a medias.

3. Recuento de leyes, ley 27 e hipótesis de la parte 19

El recuento de leyes, cerrada la parte 18.

```
ley 13  lo que no se mira deja de funcionar en silencio       50
ley 15  la señal existe y nadie la mira                       39
ley 22  un procedimiento nunca ejecutado no funciona          34
ley 14  el coste se decide al crear, no al pagar              31
ley 16  un control que estorba se rodea                       29
ley 20  lo que no tiene dueño se filtra y se desperdicia      28
ley 21  el acoplamiento vive en quién escribe                 23
ley 25  lo provisional sobrevive a su motivo                  18
ley 23  la capacidad la limita lo que ya se mantiene          16
ley 26  el valor por defecto sirve a la demostración          12
ley 24  lo que no está en el diagrama no se analiza           12
ley 17  se optimiza la medida, no el objetivo                 12
ley 19  la compensación hace invisible el fallo               10
ley 18  lo asíncrono traslada la garantía, no la elimina       8
```

Y la parte 18 obliga a escribir una ley nueva, que explica el fallo de la predicción 2:

```
LEY 27
  un control solo actúa sobre lo que cambia;
  lo que ya existe sigue incumpliendo hasta que alguien
  lo toque

apariciones en esta parte                                    5
  clase 217   mover 61 suscripciones al grupo correcto
              corrigió CERO recursos existentes; 14.200
              siguieron incumpliendo
  clase 219   la automatización de zonas DNS no enlazó las
              creadas antes; 39 radios quedaron fuera
  clase 220   el despliegue incremental nunca borra lo que
              deja de estar declarado: 1.140 huérfanos
  clase 222   activar la política de red no cambia lo ya
              desplegado, y en algunas configuraciones
              exige recrear el clúster
  clase 226   cerrar una recomendación no impide que el
              recurso siguiente nazca igual

y lo que la distingue
  la ley 25 dice que lo provisional se queda
  la ley 26 dice que el valor inicial está mal elegido
  la 27 dice algo distinto: que ARREGLAR LA REGLA no
  arregla lo que ya se hizo con la regla anterior

  → y el remedio no es una política mejor: es una TAREA DE
    REMEDIACIÓN, que es trabajo, y hay que planificarlo
  → toda implantación de un control necesita dos planes:
    el de lo nuevo y el de lo viejo
```

La hipótesis de la parte 19 (clases 229 a 240, Google Cloud en producción), escrita antes de estudiarla:

```
1. la tercera nube confirmará la equivalencia conceptual y
   volverá a diferir en el modelo operativo; y esta vez lo
   que más costará no será aprender lo distinto sino
   DESAPRENDER lo de las dos anteriores: los errores serán
   de trasladar suposiciones que aquí no valen

2. la jerarquía de proyectos y la red compartida serán otra
   vez la decisión que condiciona todo, con un matiz: aquí
   el proyecto es una frontera más fuerte que la suscripción,
   y eso hará que el error frecuente sea el CONTRARIO —
   demasiados proyectos, no demasiado pocos

3. la identidad volverá a ser el eje, y el error más
   frecuente volverá a ser el ámbito amplio: una asignación
   en la organización o en la carpeta en vez de en el
   recurso                                      ley 27, 218

4. los valores por defecto volverán a estar mal elegidos para
   producción, y aquí fallarán sobre todo en RED: la red
   global y los servicios con acceso público por defecto
   serán la fuente principal                      ley 26

5. y la predicción que ya no tiene mérito, con su corolario:
   los problemas del proyecto serán otra vez de las leyes
   25, 15 y 22. Lo que sí es refutable es esto: **la
   proporción de hallazgos que detecte una comprobación
   automática y periódica será mayor que en las partes 17 y
   18**, porque en esas dos aprendimos a montarlas. Si no lo
   es, el aprendizaje no se está trasladando
```

Y el cierre de la parte 18: de once clases, lo que más dinero movió no fue ninguna decisión de arquitectura sino dos casillas de un asistente de creación —el nivel de consistencia y la política de índices—, y lo que más riesgo concentró fue una sola asignación de permisos con el ámbito demasiado alto. La parte 19 hace el mismo recorrido en Google Cloud, empezando por su jerarquía de recursos y su red compartida. Es la clase 229.

4. Comparar las dos nubes, con cifras

Con el mismo sistema montado en dos nubes, se puede comparar de verdad. Y conviene, porque es la única forma de separar lo que es propio del proveedor de lo que es propio del método.

```
LO QUE RESULTÓ EQUIVALENTE
```

```
  el techo de disponibilidad se calcula igual   clase 185
  la aritmética de colas y concurrencia         clase 186
  la decisión de consistencia por operación     clase 187
  la idempotencia y la cola de fallidos    clases 210, 224
  la disciplina de despliegue escalonado
  la estructura del código por ciclo de vida
  y las pruebas negativas, una a una

LO QUE NO
  el modelo de gobierno: en Azure las políticas se heredan
    y son fuertes; en AWS las barreras y los controles
    están más repartidos
  el modelo de permisos: papel más ámbito frente a política
    con condiciones
  la observabilidad: aquí hay que crear el diagnóstico
    recurso a recurso
  y la consistencia configurable, que no tiene equivalente
```

Y la comparación de esfuerzo, medida:

```
                              AWS         Azure
tiempo hasta el primer
  despliegue productivo      9 semanas    7 semanas
  → más rápido la segunda vez, por el método, no por la
    nube

líneas de infraestructura
  como código                ~4.100      ~3.800

valores por defecto que hubo
  que cambiar                    24          19

pruebas negativas ejecutadas     19          19
  que fallaron la primera vez     7           8

coste mensual del mismo sistema  16.700 €   17.400 €
  → diferencia del 4 %, dentro del ruido

tiempo dedicado a aprender el
  modelo OPERATIVO               —      3 meses
  → y esto es lo que la predicción 4 acertaba
```

Y las dos conclusiones que se pueden defender:

```
1  el coste de un sistema equivalente es parecido
   → elegir nube por precio de lista es un error
   → lo que cambia el coste son las decisiones de diseño,
     no el proveedor                      clases 216, 227

2  el coste de OPERAR dos nubes no es el doble: es más
   dos modelos de gobierno, dos de permisos, dos de
   observabilidad, dos calendarios de actualización
   → y por eso la multinube se justifica por requisitos,
     no por preferencia                     clase 157
```

Y una advertencia sobre el traslado:

```
las suposiciones de la nube anterior son el mayor riesgo
  «esto se hereda» — aquí no
  «esto ya viene activado» — aquí no
  «el borrado no se propaga» — aquí sí
→ y por eso el orden de la parte empieza otra vez por lo
  que condiciona todo, en vez de por lo que ya se sabe
```

Y la lista de comprobación de la clase:

- ■ el sistema se construyó en el orden por coste de cambio
- ■ hay lista explícita de valores por defecto cambiados
- ■ el alcance de cada identidad está medido
- ■ las zonas DNS están enlazadas a todas las redes
- ■ la consistencia está decidida por operación
- ■ la instrumentación es estándar
- ■ se simulan técnicas y se publica la proporción detectada
- ■ hay plan de remediación de lo existente, no solo de lo nuevo

- las 19 pruebas negativas se ejecutaron y hay fallos publicados
- el entorno se despliega de cero desde el código
- el coste está atribuido por encima del 90 %
- está escrito lo que no se hace

Y el cierre que enlaza con la clase siguiente: la parte 19 repite el recorrido en Google Cloud, con el riesgo añadido de trasladar suposiciones que allí no valen. Empieza por la jerarquía de recursos y la red compartida, en la clase 229.

Ejemplo trabajado

El sistema de CloudShop en producción en Azure. Lo que sigue es la lista de valores por defecto cambiados, el resultado de las diecinueve pruebas negativas —de las que fallaron ocho— y la comparación con el mismo sistema en AWS.

La lista de valores por defecto cambiados:

servicio	por defecto	cambiado a
base distribuida	consistencia fuerte	acotada 5 s, sesión y eventual por petición
base distribuida	indexa todos los campos	7 de 61
base distribuida	1 región	2 (lectura)
grupo de seguridad	salida a internet permitida	ruta al cortafuegos
recursos	sin diagnóstico	por política
registros	retención global	por tabla
registros	plan analítico	básico donde procede
app service	acceso público	punto privado
app service	salida pública	integración de red
app service	config no específica de ranura	marcada por ranura
clúster	política de red desactivada	activada
clúster	red plana (asistente)	superpuesta
bus	bloqueo 30 s	120-300 s
bus	sin renovación de bloqueo	activada
flujo	retención 1 día	3 días
flujo	posición antes de procesar	después
despliegue	modo incremental sin ciclo de vida	pilas de despliegue
políticas	asignadas en suscripción	en grupos de administración
alertas	grupos por equipo	uno por turno
total		19
que funcionaban sin cambiarlos		19

Las diecinueve pruebas negativas: ocho fallaron.

✓	quitar una política heredada desde suscripción	imposible
✓	crear recurso sin etiquetas	rechazado
✓	crear en región no permitida	rechazado
✗	obtener la identidad de otro espacio de nombres → 1 de 7 credenciales federadas ataba solo el emisor: la del operador de base vectorial, añadida después de la revisión	clase 222
✓	asumir identidad federada desde otro repositorio	denegado
✗	activar papel privilegiado sin aprobación → 2 papeles se habían añadido como elegibles sin exigir aprobación, al crear un equipo nuevo	
✓	entrar con la cuenta de emergencia	45 s
✓	alcanzar un radio desde otro radio	denegado
✗	resolver nombre con punto privado desde cada red → 2 radios creados el mes anterior no tenían las zonas enlazadas: la automatización se había desplegado DESPUÉS de crearlos	ley 27, 219

```
✓  alcanzar servicio con punto privado desde internet
                                                  denegado
✗  sacar datos a un destino no declarado
   → funcionó desde el clúster: la política de red
     restringía entre pods, no la salida a internet
                                               clase 222
✓  desplegar quitando un recurso de la plantilla   borrado
✗  borrar la pila y comprobar los datos
   → una cuenta de almacenamiento de un servicio nuevo no
     tenía bloqueo de borrado ni estaba en la pila de datos
✓  mismo mensaje 50 veces                          1 efecto
✓  mensaje en cola de fallidos → alerta             38 s
✗  perder una zona completa
   → 12 min 40 s de degradación: dos servicios añadidos
     tras el último experimento no tenían restricción de
     reparto                                   clase 227
✗  simular 14 técnicas de ataque                  13 de 14
   → la que falta, aceptada por escrito
✗  provocar la condición de cada alerta
   → 4 de 58 no llegaron; 3 por umbral y 1 en estado de
     error                                     clase 225
✓  desplegar el entorno de cero                    94 min
```

Y el análisis de las ocho:

```
seis por elementos AÑADIDOS después de la última revisión
  · una credencial federada
  · dos papeles elegibles
  · dos radios sin zonas enlazadas
  · una cuenta de almacenamiento sin bloqueo
  · dos servicios sin restricción de reparto
dos por un control que no cubría lo que se creía
  · política de red que no restringe la salida
  · umbrales de alerta

→ y es el mismo diagnóstico de la clase 216: el sistema
  creció y las comprobaciones no crecieron con él
→ con un matiz nuevo: cuatro de los seis casos son de la
  ley 27, porque la automatización llegó después que el
  recurso
```

Y la corrección de método que salió:

```
toda automatización nueva se acompaña de una TAREA DE
REMEDIACIÓN sobre lo existente
y una comprobación periódica que compare el estado real
con el esperado, no solo en el momento de crear

→ y esa comprobación encontró, en los tres meses
  siguientes
    5 radios sin zonas enlazadas
    3 recursos sin bloqueo de borrado
    2 credenciales federadas mal atadas
  → todos, creados después de la automatización
    correspondiente
```

Las cifras del sistema en producción, tras tres meses:

```
p99 del flujo de compra                    < 500 ms    438 ms
disponibilidad observada                     99,8 %    99,84 %
coste mensual                             15.000 €   17.400 €
coste por pedido                            0,050 €    0,051 €
pérdida de una zona: degradación             0 min    12 min 40
  (tras corregir)                                        0 min
coste atribuido                               90 %      94 %
alertas por turno                             < 2        0,9
técnicas simuladas detectadas                 > 90 %    13/14
despliegue del entorno de cero                  —       94 min
```

La comparación con AWS, del mismo sistema:

```
p99 del flujo de compra                   412 ms      438 ms
disponibilidad observada                  99,86 %     99,84 %
coste mensual                          16.700 €    17.400 €
coste por pedido                        0,046 €     0,051 €
```

```
valores por defecto cambiados                24          19
pruebas negativas fallidas                    7           8
tiempo hasta producción                9 semanas   7 semanas
líneas de infraestructura                ~4.100      ~3.800
equipo dedicado a la plataforma            2,1         2,4
  → la diferencia está en el modelo operativo, no en la
    tecnología
```

Y las tres cosas que se decidió no hacer:

```
no operar las dos nubes en activo-activo: el coste de
  operar dos modelos de gobierno no compensa   clase 157
no migrar las cargas de AWS: funcionan
no unificar la observabilidad en un solo proveedor
  todavía: se revisa cuando el volumen lo justifique
```

La lección que este proyecto deja: los diecinueve valores por defecto cambiados funcionaban todos sin cambiarlos, y dos de ellos —dos casillas de un asistente— costaban dos mil setecientos euros al mes. De las ocho pruebas negativas fallidas, cuatro fueron por recursos creados antes de que existiera la automatización que los cubriría, que es la ley 27 en su forma más directa. Y comparado con AWS, el mismo sistema costó un 4 % más y necesitó un 14 % más de equipo: la diferencia está en operar dos modelos, no en la tecnología.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-18-azure-production-architecture/228-proyecto-cloudshop-productivo-en-azure/lab.py
```

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

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

Evidencia esperada

El artefacto principal es un incremento integrado, demostrable y documentado. Además, la entrega debe incluir el comando ejecutado, la salida estructurada y una conclusión que no exceda lo observado.

Reto verificable

Construye cloudshop-azure 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 implanta un control y los recursos existentes siguen incumpliendo	Los controles actúan sobre creaciones y modificaciones, no hacia atrás	Toda implantación necesita dos planes: el de lo nuevo y una tarea de remediación de lo existente.
Una automatización cubre unos recursos y no otros	Los recursos anteriores a la automatización no pasaron por ella	Ejecuta la automatización sobre lo existente y añade una comprobación periódica que compare el estado real con el esperado.
Al trasladar la arquitectura a otra nube fallan cosas que en la anterior funcionaban	Se trasladaron suposiciones del modelo operativo anterior	Empieza por lo que condiciona todo en la nube nueva y comprueba explícitamente qué se hereda, qué viene activado y qué se propaga.
Las pruebas negativas pasan y aparecen huecos nuevos	El sistema creció y las comprobaciones no crecieron con él	Añade una prueba negativa por cada capacidad nueva y ejecútalas periódicamente, no solo al terminar.
Se elige la nube por precio de lista y no se ahorra	El coste lo deciden las decisiones de diseño, no el proveedor	Compara el coste del mismo sistema montado bien en ambas; la diferencia suele estar dentro del ruido.
Operar dos nubes consume mucho más equipo del previsto	Se contó el código y no los modelos de gobierno, permisos, observabilidad y actualizaciones	Justifica la multinube por requisitos y no por preferencia, contando el coste operativo real.

Seguridad, ética y costo

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

Preguntas de comprobación

1. ¿Cuál de las cinco predicciones de la clase 216 falló y en qué mitad?
2. ¿Qué dice la ley 27 y en qué se distingue de las leyes 25 y 26?
3. ¿Por qué saber las leyes 25, 15 y 22 no basta para evitar que se cumplan?
4. ¿Qué se correspondió entre las dos nubes y qué no?
5. ¿Cuántas de las pruebas negativas fallidas se deben a la ley 27?

Referencias

- Microsoft (2025). Azure Well-Architected Framework. <https://learn.microsoft.com/en-us/azure/well-architected/>
- Microsoft (2025). Cloud Adoption Framework: Azure landing zones. <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/landing-zone/>
- Microsoft (2025). Azure Policy remediation tasks. <https://learn.microsoft.com/en-us/azure/governance/policy/how-to/remediate-resources>
- Beyer, B. y otros (2018). The Site Reliability Workbook. <https://sre.google/workbook/table-of-contents/>
- Basiri, A. y otros (2016). Chaos engineering. <https://ieeexplore.ieee.org/document/7503833>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 18 en PDF · Recorrido de Azure en PDF · Manual integral

Anterior	Índice	Siguiente
← 227 · Cost Management, Advisor, resiliencia y Chaos Studio	Parte 18 · Programa	229 · Resource Manager, folders, Shared VPC y guardrails →

Evaluación — 228 Proyecto: CloudShop productivo en Azure

Preguntas

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

Reto verificable

Entrega cloudshop-azure 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.