Multi-Cloud Engineering Program

Parte 19 — Google Cloud: arquitectura de datos 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	19 de 23
Clases	229–240
Evaluaciones	12
Horas estimadas	52

Índice

Parte 19 — Google Cloud: arquitectura de datos y operación en producción

← Parte 18 · Índice completo · Parte 20 →

Descargar: Esta parte en PDF · Recorrido de Google Cloud en PDF · Manual integral

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

Operar CloudShop con la red global, servicios administrados, seguridad y plataforma de datos de Google Cloud.

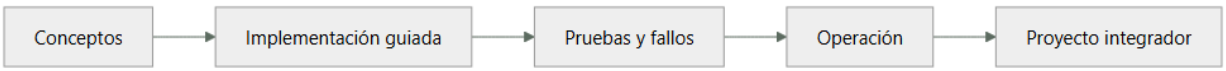
Resultados de la parte

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

Google Cloud combina una jerarquía estricta de recursos con una red global y servicios data-first; proyecto, cuota e IAM forman una unidad operativa.

Secuencia



Clases

ID	Clase	Laboratorio	Horas
229	Resource Manager, folders, Shared VPC y guardrails	governance	4
230	Workload Identity Federation, IAM Conditions y PAM	iam	4
231	Red global, load balancing, PSC y Cloud DNS	network	4
232	Terraform, Infrastructure Manager y policy validation	iac	4
233	Cloud Run, Functions, API Gateway y Workflows	serverless	4
234	GKE Autopilot, Workload Identity y Config Sync	kubernetes	4
235	Cloud SQL, Spanner, Firestore y Bigtable	data	4
236	BigQuery, Dataflow, Dataproc y gobernanza de datos	data	4
237	Pub/Sub, Eventarc y entrega exactamente-una-vez	messaging	4
238	Cloud Operations, Trace y OpenTelemetry	observability	4
239	SCC, VPC Service Controls, KMS y FinOps	security	4
240	Proyecto: CloudShop productivo en Google Cloud	capstone	8

Evaluación de la parte

- 35 % laboratorios y evidencia reproducible.
- 25 % retos verificables.
- 20 % decisiones y ADR.
- 10 % seguridad, confiabilidad y costo.

- 10 % proyecto integrador de la parte.

Se aprueba con 80 % de clases en nivel B o superior y el proyecto integrador reproducible.

Pauta bibliográfica

- Google Cloud Architecture Framework — Google.
- Google Cloud Certified Professional Cloud Architect Study Guide — Dan Sullivan.
- Data Engineering with Google Cloud — Adi Wijaya.

229 — Resource Manager, folders, Shared VPC y guardrails

Parte: 19 — Google Cloud: arquitectura de datos y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: governance · Estado: EXECUTABLECORE

Propósito

Diseñar la jerarquía de Google Cloud —organización, carpetas y proyectos— y su red compartida, con la advertencia que abre la parte: el mayor riesgo aquí no es lo que se desconoce, sino trasladar suposiciones de las dos nubes anteriores. La clase explica por qué el proyecto es una frontera más fuerte que una suscripción, qué controlan las políticas de organización frente a los permisos, y por qué la red compartida separa quién opera la red de quién la usa.

Resultados de aprendizaje

Al finalizar podrás:

1. Repartir cargas entre carpetas y proyectos con criterio.
2. Distinguir política de organización de permiso, y usar cada una.
3. Diseñar red compartida separando operación de uso.
4. Evitar los errores de trasladar suposiciones de otras nubes.
5. Corregir una jerarquía con demasiados o muy pocos proyectos.

Conceptos centrales

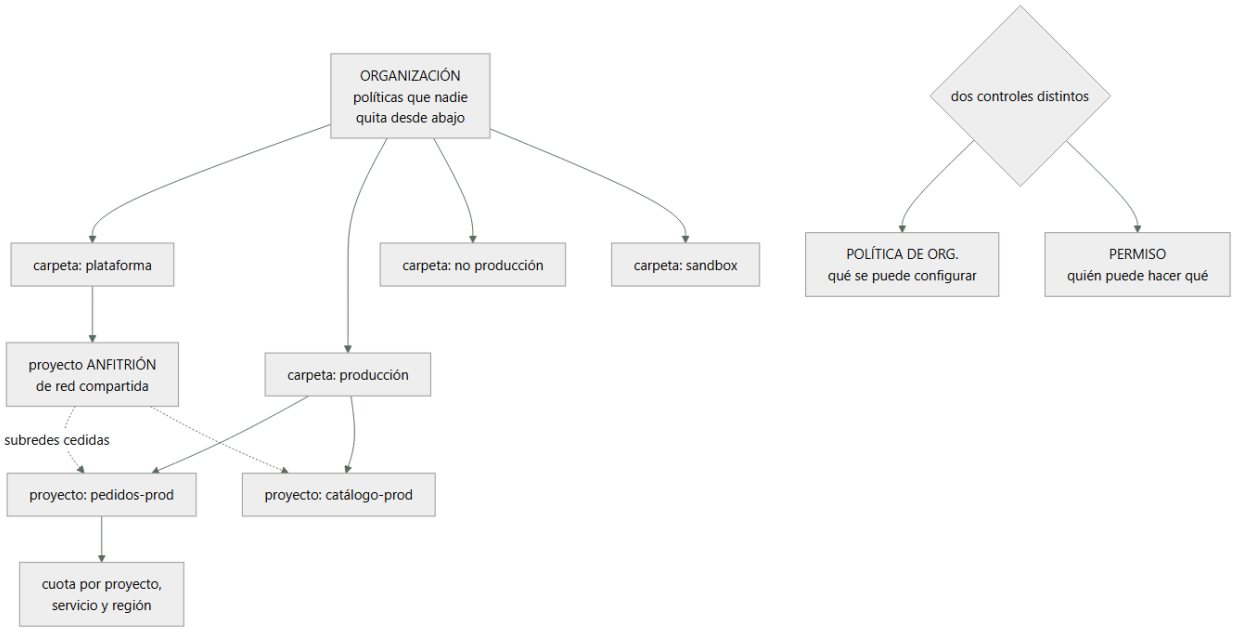
Concepto	Comprensión verificable
organización	Raíz de la jerarquía, ligada al dominio. Donde se aplican las políticas que nadie puede quitar desde abajo.
carpeta	Agrupación de proyectos y otras carpetas. Nivel intermedio donde se heredan políticas y permisos.
proyecto	Frontera de recursos, facturación, cuota, API habilitadas y red. Más fuerte que una suscripción.
política de organización	Restricción sobre qué se puede configurar. Distinta de un permiso: limita el qué, no el quién.
red compartida	Red de un proyecto anfitrión, usada por proyectos de servicio que no la administran.
cuota	Límite por proyecto, servicio y región. Se topa antes que la capacidad y se pide con antelación.

Modelo mental

Google Cloud combina una jerarquía estricta de recursos con una red global y servicios data-first; proyecto, cuota e IAM forman una unidad operativa.

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 el proyecto es fuerte

La estructura tiene tres niveles y el del medio es opcional, pero el de abajo tiene más peso que sus equivalentes en otras nubes.

```
ORGANIZACIÓN
  raíz, ligada al dominio
  aquí van las políticas de organización y los permisos que
  se heredan

CARPETA
  agrupación de proyectos, anidable
  nivel natural para políticas por entorno o por unidad

PROYECTO
  frontera de
    recursos
    facturación
    CUOTAS
    API habilitadas          ← esto no existe en otras nubes
    red (salvo que use una compartida)
    y ámbito natural de muchos permisos
```

Y la diferencia que hay que interiorizar:

```
UN PROYECTO ES MÁS BARATO Y MÁS FUERTE que una suscripción
  crear uno es inmediato y gratis
  las API se habilitan por proyecto: lo que no está
    habilitado, no existe
  y borrar un proyecto se lleva todo lo de dentro

→ y por eso el error frecuente aquí es el CONTRARIO al de
  Azure: demasiados proyectos, no demasiado pocos
```

Y el criterio de reparto, con la tensión que hay que resolver:

```
UN PROYECTO POR CARGA Y ENTORNO
  + aislamiento fuerte, cuota propia, facturación clara
  – más proyectos que gestionar: políticas, permisos,
    presupuestos, conexiones de red

UN PROYECTO POR EQUIPO Y ENTORNO
  + menos objetos que gestionar
  – las cargas comparten cuota y se afectan

CRITERIO
```

```
  por carga y entorno para producción
  agrupado para desarrollo, si el volumen no lo justifica
  → y con la entrega AUTOMATIZADA, para que crear uno cueste
    minutos y no semanas                          ley 16
```

Y las carpetas, que casi siempre se organizan de una de dos formas:

```
POR ENTORNO         producción · no producción · sandbox
  + las políticas por entorno son las que más varían
  − las unidades de negocio quedan mezcladas

POR UNIDAD, y dentro por entorno
  + refleja la organización
  − la empresa se reorganiza antes que la nube  clase 183

→ por entorno arriba y por unidad debajo suele envejecer
  mejor
```

Y una advertencia de traslado, la primera de muchas:

```
✗ «esto se hereda como allí»
  las políticas de organización se heredan, sí
  pero su combinación tiene reglas propias: algunas se
  fusionan y otras se sustituyen
  → hay que leerlas, no suponerlas
```

2. Políticas de organización y permisos: dos cosas

Aquí hay una separación que en otras nubes está mezclada, y confundirla produce controles que no controlan.

```
POLÍTICA DE ORGANIZACIÓN
  restringe QUÉ se puede configurar
  ejemplos
    prohibir direcciones IP externas en máquinas
    restringir las regiones donde se puede crear
    prohibir claves de cuenta de servicio
    exigir que los recursos no sean públicos
    restringir qué dominios pueden recibir permisos
  → se aplica en organización, carpeta o proyecto
  → y NO depende de quién lo intente: ni el dueño puede

PERMISO
  decide QUIÉN puede hacer QUÉ sobre QUÉ recurso
  → y se hereda hacia abajo

→ la política de organización es la barrera; el permiso es
  la puerta
→ y una barrera bien puesta hace que el permiso amplio sea
  menos peligroso                              clase 169
```

Y las políticas que conviene tener desde el primer día:

```
prohibir claves de cuenta de servicio
  → obliga a usar federación e identidad de carga
                                                clase 230
prohibir IP externas salvo donde se declare
restringir regiones
prohibir que los recursos se hagan públicos
restringir a qué dominios se pueden dar permisos
  → impide dar acceso a una cuenta personal por error
prohibir el reenvío de puertos y el acceso directo
exigir el uso de la red compartida
```

Y la disciplina de implantación, que es la de siempre:

```
1  aplicar en modo de solo registro donde exista
2  medir el incumplimiento actual
3  corregir o exceptuar, con dueño y fecha
4  aplicar
→ y recordando que no actúa hacia atrás           ley 27
```

Las API habilitadas, que son un control que sorprende:

```
un servicio no se puede usar si su API no está habilitada
en el proyecto
→ es un control de superficie muy efectivo y gratuito
→ habilitar solo lo que se usa reduce el alcance de un
```

```
  compromiso                                  clase 189

y la trampa
  habilitar una API es fácil y nadie la deshabilita
  → inventario periódico de API habilitadas y sin uso
                                                  ley 25
```

Y las cuotas, con la lección de la clase 222:

```
son por proyecto, servicio y región
se topan antes que la capacidad
se piden con antelación y tardan
→ margen del 30 % sobre el máximo del escalado
→ alerta al 80 %                              clase 262
→ y no actualizar ni escalar si el margen no da
```

3. Red compartida: separar operar de usar

La red compartida es la pieza que más distingue a esta nube en gobierno, y resuelve un problema real.

```
EL PROBLEMA
  si cada proyecto tiene su red, hay que conectarlas todas
  entre sí
  → y cada equipo administra su red, con lo que eso implica

LA RED COMPARTIDA
  un PROYECTO ANFITRIÓN contiene la red y sus subredes
  los PROYECTOS DE SERVICIO usan esas subredes
  → los recursos viven en el proyecto de servicio
  → la red la administra el equipo de plataforma

QUÉ SEPARA
  quién OPERA la red: plataforma
  quién la USA: los equipos
  → y un equipo no puede crear rutas, emparejamientos ni
    reglas por su cuenta
```

Y las decisiones que hay que tomar:

```
¿UNA RED COMPARTIDA O VARIAS?
  una por entorno es lo habitual
  → producción y no producción NUNCA en la misma

¿QUÉ SUBREDES SE CEDEN A QUIÉN?
  se ceden subredes concretas a proyectos concretos
  → un proyecto de servicio solo ve las subredes cedidas
  → y eso es un control de alcance real       clase 199

¿QUIÉN PUEDE CREAR REGLAS DE CORTAFUEGOS?
  plataforma, con etiquetas o cuentas de servicio como
  selector
  → y los equipos piden, no crean
  → si pedir tarda días, crearán recursos con IP pública
                                                  ley 16
```

Y las particularidades de red que hay que conocer para no trasladar suposiciones:

```
LA RED ES GLOBAL, las subredes son regionales
  → una sola red puede abarcar todas las regiones
  → y por tanto no hace falta emparejar entre regiones
  → es distinto de las otras dos nubes            clase 231

LAS REGLAS DE CORTAFUEGOS SON DE LA RED, no de la subred
  con prioridades y con selectores por etiqueta o por
  cuenta de servicio
  → seleccionar por cuenta de servicio es mucho mejor que
    por etiqueta: la etiqueta la puede poner cualquiera con
    permiso de instancia

LA SALIDA A INTERNET ESTÁ PERMITIDA por defecto
  → y hay que cerrarla explícitamente             ley 26
```

Y el error de traslado más caro en esta parte:

```
✗ «cada proyecto tendrá su red, como cada cuenta la suya»
  → produce decenas de redes que hay que conectar
```

```
  → y pierde la ventaja principal de esta nube
→ la red compartida es la elección por defecto salvo razón
  concreta
```

4. Facturación, entrega y corrección

La facturación, que se organiza distinto:

```
una CUENTA DE FACTURACIÓN se asocia a muchos proyectos
  → y el desglose por proyecto es directo y limpio
  → por eso la separación por proyecto es también una
    decisión de atribución                    clase 239

y las etiquetas
  aquí se llaman de otra forma y hay dos mecanismos
    unas sirven para facturación y organización
    otras para condiciones de permiso y políticas
  → y no son intercambiables: hay que saber cuál se usa
    para qué
```

La entrega de proyectos, que decide si el gobierno se cumple:

```
UN PROYECTO ENTREGADO trae
  colocado en la carpeta correcta
  cuenta de facturación asociada
  API habilitadas mínimas
  subredes cedidas de la red compartida
  permisos base y cuentas de servicio
  destinos de registro configurados          clase 238
  presupuesto y alertas
  y las políticas heredadas aplicando

Y EL PLAZO
  si tarda semanas, los equipos usarán proyectos
  personales o pedirán permisos amplios en los que ya
  tienen                                        ley 16
  → automatizar la entrega es lo que hace cumplir el resto
```

Corregir una jerarquía, con el problema propio de esta nube:

```
DEMASIADOS PROYECTOS
  el síntoma: cientos de proyectos, muchos sin uso
  → y cada uno con su presupuesto, permisos y API
  corrección
    inventario con último uso y dueño
    los sin actividad en 90 días, a retirar     ley 25
    y consolidar los de desarrollo por equipo

PROYECTOS EN LA CARPETA EQUIVOCADA
  mover es sencillo y las políticas empiezan a aplicar
  → pero lo ya creado sigue incumpliendo         ley 27
  → hace falta remediación, no solo mover

PROYECTOS SIN CARPETA, colgando de la organización
  → no reciben las políticas por entorno
  → es el hallazgo típico del primer inventario
```

Y lo que hay que vigilar:

```
proyectos sin carpeta, sin dueño o sin presupuesto
API habilitadas y sin uso
cuotas frente al máximo del escalado
políticas de organización con excepciones, y su antigüedad
subredes cedidas y no usadas
y recursos creados fuera de la red compartida
```

Y la lista de comprobación de la clase:

- ■ hay carpetas por entorno, y proyectos por carga
- ■ ningún proyecto cuelga directamente de la organización
- ■ las políticas de organización están en organización o carpeta
- ■ están las políticas mínimas: sin claves de cuenta de servicio, sin IP externas, regiones, no público, dominios
- ■ se midió el incumplimiento antes de aplicar
- ■ hay plan de remediación de lo existente

- solo están habilitadas las API que se usan
- hay red compartida por entorno, con subredes cedidas
- las reglas de cortafuegos seleccionan por cuenta de servicio, no por etiqueta
- la salida a internet está cerrada explícitamente
- las cuotas tienen margen y alerta al 80 %
- la entrega de un proyecto está automatizada y tarda minutos
- hay inventario de proyectos con último uso y dueño

Y el cierre que enlaza con la clase siguiente: con la jerarquía y la red repartidas, el control que decide el alcance real vuelve a ser la identidad, y aquí hay un elemento que no existe en las otras dos nubes: las condiciones en las asignaciones. Es la materia de la clase 230.

Ejemplo trabajado

CloudShop monta su organización en Google Cloud, con la experiencia de las dos nubes anteriores. Lo que sigue son los tres errores de traslado que cometió el equipo, el inventario que encontró 214 proyectos, y la red compartida que redujo 41 redes a dos.

El punto de partida, tras dos años de uso sin gobierno:

```
proyectos                                        214
  colgando de la organización, sin carpeta       189
  con dueño identificable                         94
  sin actividad en 90 días                        71
  sin presupuesto asociado                       203

redes                                             41
  una por proyecto que la necesitaba
  emparejamientos entre ellas                     67

políticas de organización aplicadas                2
claves de cuenta de servicio                     341
  con más de 1 año                               218
```

Los tres errores de traslado, cometidos en las primeras semanas:

```
ERROR 1 · «cada proyecto con su red»
  el equipo venía de un modelo de cuenta por carga con red
  propia
  creó 12 redes nuevas antes de que alguien preguntara
  → y la conexión entre ellas exigía emparejamientos, que
    no son transitivos

  al descubrir la red compartida
    12 redes nuevas → 0
    41 redes existentes → 2 (producción y no producción)
    67 emparejamientos → 4
  y una diferencia que no habían previsto
    la red es GLOBAL: una sola red cubre las 3 regiones
    → no hacen falta emparejamientos entre regiones

ERROR 2 · «las reglas de cortafuegos son por subred»
  se crearon reglas asumiendo que aplicaban a la subred
  → aquí son de la RED, con selectores
  → y las primeras se escribieron con selectores por
    ETIQUETA
  → cualquiera con permiso sobre una instancia podía
    ponerle la etiqueta que abría el acceso a la base

  corrección
    selectores por CUENTA DE SERVICIO
    → poner una cuenta de servicio a una instancia exige
      permiso sobre esa cuenta
    → y eso sí es un control                    clase 230

ERROR 3 · «los permisos amplios ya los limita la barrera»
  se asumió el modelo de la nube anterior
  → aquí la política de organización limita QUÉ se
    configura, no QUIÉN accede a los datos
  → una asignación amplia sobre almacenamiento seguía
```

```
    dando acceso a los datos, con la barrera puesta

  corrección
    los dos controles, y no uno en lugar del otro
                                                clase 230
```

Y la conclusión que el equipo escribió:

```
los tres errores fueron de SUPOSICIÓN, no de
desconocimiento
cada uno venía de dar por válido algo de otra nube
→ y los tres se detectaron por casualidad, no por una
  comprobación
→ desde entonces, la primera tarea de cada servicio nuevo
  es leer qué se hereda, qué viene activado y qué se
  propaga                                    clase 228
```

La jerarquía nueva:

```
organización
■■■ plataforma
■    ■■■ red-compartida-prod       (proyecto anfitrión)
■    ■■■ red-compartida-nopro      (proyecto anfitrión)
■    ■■■ identidad
■    ■■■ observabilidad
■    ■■■ seguridad
■■■ produccion
■    ■■■ pedidos-prod
■    ■■■ catalogo-prod
■    ■■■ precios-prod
■    ■■■ ... (14 proyectos, uno por carga)
■■■ no-produccion
■    ■■■ pedidos-pre, pedidos-dev
■    ■■■ ... (28 proyectos)
■■■ sandbox                        con caducidad de 60 días
```

Las políticas de organización, implantadas por orden:

```
política                          incumplimiento inicial
sin claves de cuenta de servicio            341
sin IP externas en máquinas                 118
regiones restringidas a la UE                41
sin recursos públicos                        29
dominios permitidos para permisos            14
sin reenvío de puertos                       62
uso obligatorio de red compartida           n/a (nuevo)

el orden que se siguió
  1  dominios permitidos      ← 14 casos, 2 días
     y encontró 3 permisos concedidos a cuentas personales
     de ex empleados                              ley 25
  2  sin recursos públicos    ← 29 casos, 1 semana
     de ellos, 4 almacenes con datos de clientes
  3  regiones                 ← 41 casos, 3 semanas
  4  sin IP externas          ← 118 casos, 6 semanas
     sustituidas por acceso mediante servicio de
     administración                           clase 256
  5  sin claves de cuenta de servicio ← 341, 4 meses
     el más largo, y el de más valor          clase 230

y la remediación, que no vino sola
  aplicar las políticas corrigió 0 recursos existentes
  → tareas de remediación planificadas aparte    ley 27
```

El inventario de proyectos:

```
214 proyectos, con último uso y dueño

  con actividad y dueño                            94
  con actividad y SIN dueño                        49
    → 12 resultaron ser de equipos que ya no existen
    → 3 tenían cargas en producción que nadie sabía que
      existían                                  ley 20
  sin actividad en 90 días                         71
    → apagados, y borrados 30 días después
```

```
    → 2 reclamaciones, ambas restauradas en minutos

proyectos tras la limpieza                          143
  → y consolidando los de desarrollo por equipo:     86

coste liberado                               4.900 €/mes
  de los cuales, en los 71 sin actividad      3.100 €
```

Y el hallazgo más incómodo:

```
uno de los 71 sin actividad tenía
  una base de datos con una copia de la tabla de clientes
  de 2023
  sin cifrado con clave propia
  sin registro de acceso
  y con permisos de lectura para «todos los usuarios
  autenticados» ← es decir, cualquier cuenta de Google

llevaba así                                  19 meses
lo encontró                          el inventario, no una
                                     alerta      ley 15
```

La red compartida, montada:

```
dos proyectos anfitriones: producción y no producción
red global por entorno; subredes por región
subredes cedidas a proyectos concretos
  → pedidos-prod ve su subred y nada más

reglas de cortafuegos
  denegación por defecto en entrada y en SALIDA
  selectores por cuenta de servicio
  creadas por plataforma, pedidas por los equipos
  → y el plazo de una petición: 40 minutos, automatizado
    con revisión

resultado
  redes                                  41 → 2
  emparejamientos                        67 → 4
  reglas de cortafuegos                 940 → 118
  equipos que pueden crear rutas          41 → 1
```

La entrega de proyectos, automatizada:

```
antes   petición por correo → 2 semanas
        y 189 proyectos creados fuera del proceso
después plantilla que crea el proyecto con
          carpeta, facturación, API mínimas, subred cedida,
          permisos base, destinos de registro, presupuesto
        plazo                                  14 min
        proyectos creados fuera del proceso en 6 meses: 0
```

El resultado:

```
proyectos                                 214          86
  sin carpeta                             189           0
  sin dueño                               120           0
  sin presupuesto                          203           0
redes                                       41           2
claves de cuenta de servicio               341          12
  (las 12, con excepción registrada)
recursos públicos                           29           0
permisos a cuentas fuera del dominio        14           0
coste mensual                          28.700 €    21.400 €
plazo de entrega de un proyecto        2 semanas      14 min
```

La lección que esta clase deja: los tres errores del arranque no fueron por desconocer esta nube, sino por dar por válidas suposiciones de las otras dos, y ninguno lo detectó una comprobación: se descubrieron por casualidad. Y el inventario de proyectos —que no requería tecnología ninguna— encontró setenta y uno sin actividad, tres con cargas de producción que nadie conocía y una copia de la tabla de clientes legible por cualquier cuenta de Google desde hacía diecinueve meses.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-19-gcp-production-architecture/229-resource-manager-folders-shared-vpc-y-guardrails/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 gcp-foundation 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 gcp-foundation 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
Aparecen decenas de redes que hay que conectar entre sí	Se trasladó el modelo de una red por cuenta o suscripción	Usa red compartida por entorno; la red es global y no hacen falta emparejamientos entre regiones.
Cualquiera puede abrir un acceso poniendo una etiqueta a su instancia	Las reglas de cortafuegos seleccionan por etiqueta	Selecciona por cuenta de servicio; asignarla exige permiso sobre esa cuenta.
Las barreras están puestas y siguen existiendo accesos amplios a los datos	Se confundió política de organización con permiso: la primera limita qué se configura, no quién accede	Usa los dos controles, no uno en lugar del otro.
Hay cientos de proyectos y nadie sabe cuáles sirven	Crear un proyecto es inmediato y nadie los retira	Inventario con último uso y dueño; apaga los inactivos, bórralos después y consolida los de desarrollo.
Mover proyectos a la carpeta correcta no corrige nada	Las políticas no actúan sobre lo ya creado	Planifica una tarea de remediación además del movimiento.
Los equipos crean recursos fuera del proceso	Conseguir un proyecto o una regla de cortafuegos tarda semanas	Automatiza la entrega de proyectos y las peticiones de reglas para que tarden minutos.

■ 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 hace que un proyecto sea una frontera más fuerte que una suscripción?
2. ¿Qué diferencia hay entre una política de organización y un permiso?
3. ¿Qué separa la red compartida y por qué importa?
4. ¿Por qué los selectores por cuenta de servicio son mejores que por etiqueta?
5. ¿Cuál es el error de reparto de proyectos característico de esta nube?

Referencias

- Google Cloud (2025). Resource hierarchy. <https://cloud.google.com/resource-manager/docs/cloud-platform-resource-hierarchy>
- Google Cloud (2025). Organization policy constraints. <https://cloud.google.com/resource-manager/docs/organization-policy/org-policy-constraints>
- Google Cloud (2025). Shared VPC overview. <https://cloud.google.com/vpc/docs/shared-vpc>
- Google Cloud (2025). VPC firewall rules and service account targets. <https://cloud.google.com/firewall/docs/firewalls>
- Google Cloud (2025). Landing zone design in Google Cloud. <https://cloud.google.com/architecture/landing-zones>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 19 en PDF · Recorrido de Google Cloud en PDF · Manual integral

Anterior	Índice	Siguiente
← 228 · Proyecto: CloudShop productivo en Azure	Parte 19 · Programa	230 · Workload Identity Federation, IAM Conditions y PAM →

Evaluación — 229 Resource Manager, folders, Shared VPC y guardrails

Preguntas

1. Explica el problema que resuelve esta capacidad sin mencionar un proveedor.
2. Identifica una frontera de responsabilidad y su propietario.
3. Compara dos alternativas mediante confiabilidad, seguridad, costo y operación.
4. ¿Qué demuestra el laboratorio y qué no puede demostrar?
5. Propón un caso límite o una prueba de fallo.

Reto verificable

Entrega gcp-foundation 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.

230 — Workload Identity Federation, IAM Conditions y PAM

Parte: 19 — Google Cloud: arquitectura de datos y operación en producción Nivel: avanzado · Horas estimadas: 4 Laboratorio: iam · Estado: EXECUTABLECORE

Propósito

Resolver la identidad en Google Cloud, donde el mecanismo tiene una pieza que no existe en las otras dos nubes: las condiciones en las asignaciones, que permiten conceder un permiso solo sobre ciertos recursos y durante cierto tiempo. La clase cubre la federación y las cuentas de servicio sin claves, la suplantación como sustituto de las credenciales guardadas, el ámbito de asignación —que vuelve a ser el error frecuente— y la elevación temporal con aprobación.

Resultados de aprendizaje

Al finalizar podrás:

1. Eliminar claves de cuenta de servicio con federación y suplantación.
2. Asignar permisos en el ámbito mínimo, con condiciones.
3. Distinguir quién actúa de qué identidad se usa, y auditarlo.
4. Configurar elevación temporal con aprobación y caducidad.
5. Detectar y reducir el alcance de las identidades más amplias.

Conceptos centrales

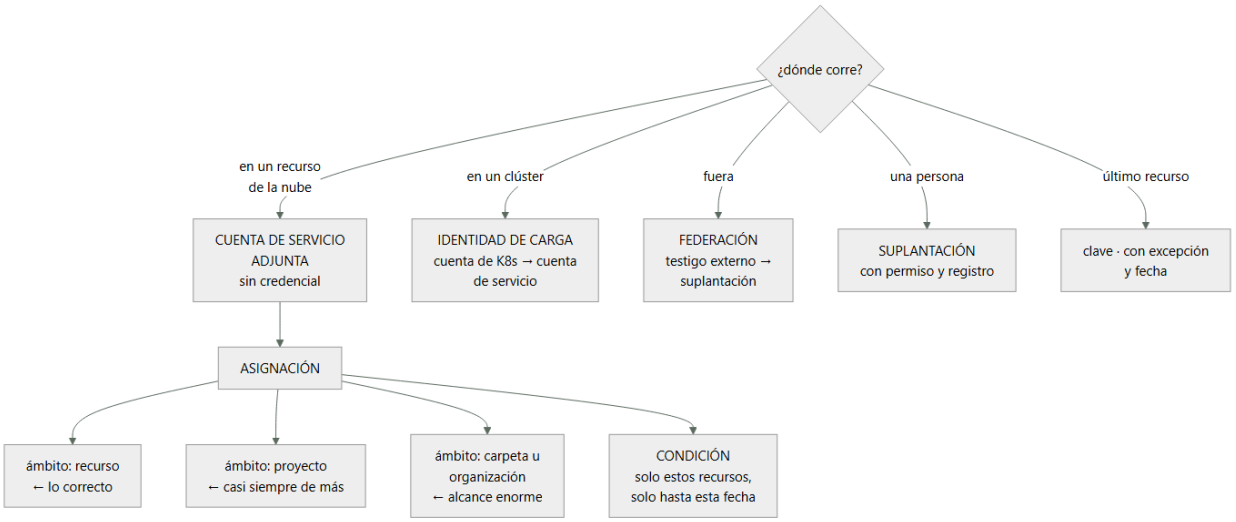
Concepto	Comprensión verificable
cuenta de servicio	Identidad de una carga. Puede usarse sin credenciales si se adjunta al recurso o se suplanta.
clave de cuenta de servicio	Credencial de larga duración en un fichero. Lo que hay que eliminar.
suplantación	Obtener un testigo de otra identidad con permiso para ello, sin credencial guardada.
federación de identidad de carga	Confianza en un emisor externo para que sus testigos suplanten una cuenta de servicio.
condición en la asignación	Expresión que limita cuándo y sobre qué recursos aplica un permiso.
identidad de carga del clúster	Vinculación entre una cuenta de servicio de Kubernetes y una de la nube, sin claves.

Modelo mental

Google Cloud combina una jerarquía estricta de recursos con una red global y servicios data-first; proyecto, cuota e IAM forman una unidad operativa.

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

■ Flujo de razonamiento



Desarrollo

1. Eliminar las claves

Una clave de cuenta de servicio es un fichero con una credencial que no caduca. Es lo que hay que eliminar, y aquí hay cuatro formas de no necesitarla.

```
1  CUENTA DE SERVICIO ADJUNTA — para lo que corre en la nube
   la máquina, la función o el servicio lleva una cuenta de
   servicio asociada
   el código obtiene testigos del servicio de metadatos
   → sin credencial en ningún sitio

   y la regla que se olvida
     NO usar la cuenta de servicio POR DEFECTO del proyecto
     → tiene permisos amplios y la comparten todos los
       recursos del proyecto
     → una cuenta por carga, con sus permisos    ley 26

2  IDENTIDAD DE CARGA — para lo que corre en un clúster
   una cuenta de servicio de Kubernetes se vincula con una
   de la nube
   → el pod obtiene testigos sin claves        clase 234

3  FEDERACIÓN — para lo que corre fuera
   se declara confianza en un emisor externo (canalización,
   otra nube, otro clúster)
   sus testigos suplantan una cuenta de servicio
   → mismo mecanismo y mismos errores que en la clase 206

4  SUPLANTACIÓN — para personas y para encadenar
   una identidad con permiso obtiene un testigo de otra
   → y el registro guarda QUIÉN suplantó a QUIÉN
```

Y la política que lo hace efectivo:

```
prohibir la creación de claves de cuenta de servicio, como
política de organización                    clase 229
→ y las existentes, inventariadas, migradas y borradas
→ con excepción registrada, dueño y fecha para lo que no se
  pueda                                       ley 25
```

La suplantación, que es la pieza que más cambia el día a día:

```
UNA PERSONA no necesita permisos de producción
  tiene permiso para SUPLANTAR una cuenta de servicio que
  sí los tiene
  → obtiene un testigo de corta duración
  → y queda registrado quién lo hizo          clase 238

Y LAS CANALIZACIONES ENCADENAN
  la identidad federada suplanta una cuenta de despliegue
```

```
  → y esa cuenta tiene los permisos, no el testigo externo
  → así se cambian los permisos sin tocar la federación
```

Y el error que hay que evitar:

```
dar a muchos el permiso de suplantar una cuenta muy
permisiva
→ es equivalente a darles esos permisos, y más difícil de
  ver en un inventario
→ el permiso de suplantar se audita como si fuera el
  permiso suplantado
```

2. Ámbito y condiciones

El error frecuente vuelve a ser el mismo de las clases 206 y 218: el ámbito demasiado amplio. Aquí hay además una herramienta que no existe en las otras dos.

```
LOS ÁMBITOS, de mayor a menor
  organización   → todos los proyectos
  carpeta        → todos los de esa rama
  proyecto       → todos sus recursos
  RECURSO        → solo ese

y la costumbre
  asignar «Editor» en el proyecto, porque funciona
  → esa identidad puede crear, modificar y borrar casi todo
  → incluida la base de datos                  clase 218
```

Las condiciones, que permiten afinar sin crear papeles a medida:

```
UNA CONDICIÓN es una expresión que limita cuándo aplica el
permiso

  POR RECURSO
    «solo sobre los almacenes cuyo nombre empiece por
     pedidos-»
    → un permiso de proyecto que en la práctica solo toca
      lo suyo

  POR TIEMPO
    «solo hasta el 31 de marzo»
    → el acceso caduca solo                       ley 25
    → y esto resuelve el problema del acceso temporal sin
      infraestructura adicional

  POR ETIQUETA
    «solo sobre recursos con entorno = desarrollo»

  POR TIPO DE RECURSO
    «solo sobre conjuntos de datos, no sobre trabajos»
```

Y el uso que más aporta:

```
el acceso concedido durante un incidente, con condición de
tiempo
→ nadie tiene que acordarse de quitarlo
→ y el registro de decisión queda en la propia asignación
```

Y las limitaciones que hay que conocer:

```
no todos los servicios soportan condiciones por recurso
  → hay que comprobarlo por servicio
las condiciones complican la auditoría: un inventario que
  no las lee muestra permisos que en realidad no aplican
  → y al revés: un permiso que parece limitado puede no
    estarlo si el servicio ignora la condición

→ comprobar con una prueba negativa, no suponer  ley 22
```

Los papeles, con la disciplina de siempre:

```
usar el papel PREDEFINIDO más específico que exista
  no «Editor» sino «Escritor de objetos de almacenamiento»
y papel PERSONALIZADO cuando ninguno encaje
  → con las acciones justas, y mantenido

y los papeles que se conceden con elevación temporal
```

```
  Propietario
  Administrador de organización
  Administrador de facturación
  Administrador de seguridad
  → nunca permanentes
```

3. Elevación temporal y auditoría

La elevación temporal aquí se puede montar de dos formas, y conviene conocer las dos.

```
CON EL SERVICIO DE GESTIÓN DE ACCESO PRIVILEGIADO
  la persona es ELEGIBLE; solicita, se aprueba, caduca
  → con justificación y registro

CON CONDICIONES DE TIEMPO
  una asignación con condición de caducidad
  → más simple, sin aprobación integrada
  → útil para accesos acordados fuera de línea

Y EN AMBOS CASOS
  activación rápida cuando no requiere aprobación
  aprobadores con turnos cuando sí                ley 16
  y un camino de emergencia sin aprobación, con alerta
    inmediata                                   clase 218
```

Y la prueba obligatoria:

```
usar el acceso de emergencia y cronometrar
→ en la clase 179 estaba creado, documentado y no
  funcionaba                                      ley 22
```

La auditoría, con lo que hay que saber leer:

```
LOS REGISTROS DE ACTIVIDAD DE ADMINISTRACIÓN
  siempre activos y sin coste
  quién creó, modificó o borró qué

LOS REGISTROS DE ACCESO A DATOS
  DESACTIVADOS por defecto, salvo unos pocos    ley 26
  → quién leyó qué
  → y son los que hacen falta para investigar una fuga
  → y los que más volumen generan: hay que elegir
    servicios y coste                          clase 238

Y EL CAMPO QUE HAY QUE MIRAR
  cuando alguien suplanta, el registro guarda la identidad
  ORIGINAL y la suplantada
  → sin leer las dos, la auditoría atribuye la acción a la
    cuenta de servicio y no a la persona
```

Medir el alcance, que es lo que ordena el trabajo:

```
el analizador de políticas responde preguntas concretas
  ¿quién puede leer este almacén?
  ¿a qué recursos puede acceder esta identidad?
  ¿quién puede suplantar esta cuenta de servicio?

→ y esas tres preguntas, ejecutadas periódicamente, son la
  medida de alcance de la clase 133

y las recomendaciones de permisos sobrantes
  la plataforma sugiere reducir papeles poco usados
  → basadas en uso real de 90 días
  → se evalúan, no se aplican a ciegas         clase 227
```

Y una fuente de alcance que se olvida:

```
LOS GRUPOS
  los permisos se conceden a grupos, y la pertenencia se
  gestiona en el directorio
  → alguien añadido a un grupo hereda permisos que nadie
    revisó
  → hay que auditar la pertenencia efectiva     clase 218
```

4. Lo que hay que comprobar

Las pruebas negativas de esta clase, que son las mismas de siempre con otras piezas:

- crear una clave de cuenta de servicio → debe fallar
- suplantar una cuenta desde una identidad sin permiso → debe fallar
- usar una identidad federada desde otro repositorio → debe fallar
- acceder a un recurso fuera de la condición → debe fallar
- usar un acceso cuya condición de tiempo ha caducado → debe fallar
- activar un papel privilegiado sin aprobación → debe fallar
- entrar con el acceso de emergencia → debe funcionar
- comprobar que el registro guarda la identidad original en una suplantación
- y que las tres preguntas de alcance devuelven lo esperado

Y las señales que hay que vigilar:

```
claves de cuenta de servicio existentes         → cero
asignaciones en organización y carpeta          → mínimas
identidades con papeles de administración
permisos concedidos y no usados en 90 días
condiciones de tiempo que vencen este mes
usos del acceso de emergencia                   → alerta
cambios en asignaciones de papeles              → auditados
suplantaciones de cuentas privilegiadas         → alerta
```

Y la disciplina de revisión:

```
trimestral
  revisión de accesos por dueño
  pertenencia efectiva de grupos
  quién puede suplantar cada cuenta privilegiada
  y prueba del acceso de emergencia

y la regla que la hace útil
  quien no responde a la revisión → se retira el acceso
                                                clase 218
```

Los errores de traslado, que esta parte vigila especialmente:

```
✗ «la barrera limita el acceso a los datos»
  la política de organización limita qué se configura
  → el acceso a los datos lo decide el permiso
                                                clase 229

✗ «con quitar la clave ya está resuelto»
  si la cuenta de servicio conserva permisos amplios, el
  problema sigue: solo cambia cómo se obtiene el testigo

✗ «la condición de recurso funciona en todos los servicios»
  → hay que comprobarlo por servicio, con una prueba
```

Y la lista de comprobación de la clase:

- la política prohíbe crear claves de cuenta de servicio
- no queda ninguna clave sin excepción registrada
- ningún recurso usa la cuenta de servicio por defecto
- lo que corre en la nube usa cuenta adjunta
- lo que corre fuera usa federación con sujeto exacto
- las personas acceden por suplantación, no con permisos propios de producción
- las asignaciones están en el ámbito del recurso
- se usan condiciones por recurso y por tiempo donde aportan
- se comprobó por servicio que la condición se respeta
- los papeles privilegiados requieren elevación temporal
- el acceso de emergencia se ha probado este trimestre
- los registros de acceso a datos están activados donde hacen falta
- la auditoría lee la identidad original en las suplantaciones

- se ejecutan las tres preguntas de alcance periódicamente

Y el cierre que enlaza con la clase siguiente: con jerarquía e identidad resueltas, queda la red, que aquí es global y tiene mecanismos de conectividad privada y de balanceo distintos de los de las otras dos nubes. Es la materia de la clase 231.

Ejemplo trabajado

CloudShop elimina las 341 claves de cuenta de servicio de su organización. Lo que sigue es el inventario, la migración por tipos, y el hallazgo de que quitar las claves no reducía el alcance.

El inventario de claves:

```
claves de cuenta de servicio                      341
  con más de 1 año                                218
  con más de 3 años                                94
  usadas en los últimos 90 días                   127
  NUNCA usadas                                     61
  usadas desde fuera de la organización            14  ←

y las 14 usadas desde fuera
  9   canalizaciones de un repositorio externo
  3   herramientas de proveedores
  1   un script en el portátil de un consultor que
      terminó su contrato en 2023            ley 25, 20
  1   uso desde una dirección que nadie pudo explicar
      → se revocó inmediatamente y se investigó
      → resultó ser un entorno de pruebas de un socio,
        no declarado
```

La migración, por tipos:

```
tipo de uso                  claves    migrado a
cargas en máquinas               88    cuenta adjunta
cargas en el clúster             64    identidad de carga
canalizaciones internas          41    federación
canalizaciones externas           9    federación
funciones y servicios            52    cuenta adjunta
scripts de personas              43    suplantación
herramientas de terceros         12    9 federadas,
                                        3 con excepción
sin uso                          32    borradas

total migrado                   341 → 3 con excepción
                                  registrada, dueño y
                                  fecha

duración                                    4 meses
```

Y los dos pasos que costaron más:

```
LAS 43 DE PERSONAS
  cada una tenía su clave para «hacer cosas rápido»
  y muchas con permisos de proyecto
  → sustituidas por suplantación de cuentas de servicio
    concretas
  → y aquí hubo resistencia: suplantar exige un comando más
  → se resolvió con un envoltorio que lo hace transparente
                                                  ley 16

LAS 88 DE MÁQUINAS
  al mirar cuál usaban, 71 usaban la CUENTA DE SERVICIO POR
  DEFECTO del proyecto
  → que tiene permisos amplios y la comparten todos los
    recursos                                      ley 26
  → no bastaba quitar la clave: había que crear una cuenta
    por carga
```

El hallazgo: quitar las claves no redujo el alcance.

```
tras migrar las 341, se midió el alcance con las tres
preguntas del analizador

  ¿a qué recursos llega cada identidad?
```

```
    identidad                  antes      después
    cuenta por defecto de
      pedidos-prod             todo el    todo el
                               proyecto   proyecto
    despliegue                 organización organización
    cuenta de análisis         41 conjuntos 41 conjuntos

  → el alcance NO había cambiado
  → solo había cambiado cómo se obtiene el testigo

y la conclusión
  eliminar credenciales resuelve el ROBO de credenciales
  no resuelve el ALCANCE
  → son dos trabajos distintos, y el segundo es el que
    reduce el daño de un compromiso        clases 189, 218
```

La reducción de alcance, con condiciones:

```
asignaciones inventariadas                        2.410
  en organización                                    18
  en carpeta                                        140
  en proyecto                                     1.870
  en recurso                                        382

las tres peores
  1  la cuenta de despliegue con «Editor» en la
     organización
     alcance: 86 proyectos, todo
     corrección: 5 cuentas, una por propósito, con
     asignación en el proyecto y CONDICIÓN por prefijo de
     nombre de recurso
     alcance: entre 4 y 40 recursos cada una

  2  el equipo de análisis con «Visor de datos» en la
     carpeta de producción
     alcance: todos los conjuntos y tablas, incluidos los
     que contienen datos personales completos
     corrección: asignación por conjunto, y para los
     sensibles, seudonimizados          clases 236, 239

  3  17 personas con «Editor» en proyectos de producción
     corrección: suplantación de cuentas concretas, con
     elevación temporal para lo privilegiado
```

Y el uso de las condiciones, con la comprobación que hizo falta:

```
se usaron condiciones en 61 asignaciones
  por prefijo de recurso                            38
  por tiempo (accesos temporales)                   19
  por etiqueta de entorno                            4

y la prueba negativa, por servicio
  se intentó acceder a un recurso FUERA de la condición
  en cada uno de los 9 servicios implicados

  respetaron la condición                            7
  NO la respetaron                                   2  ←
    → dos servicios ignoraban la condición por recurso
    → esas 2 asignaciones se sustituyeron por papeles
      personalizados y asignación en el recurso

→ y sin esa prueba, dos permisos habrían parecido
  limitados y no lo estaban                      ley 22
```

La elevación temporal:

```
papeles privilegiados retirados como permanentes
  Propietario                                  14 → 0
  Administrador de organización                 6 → 0
  Administrador de facturación                  4 → 0
  Administrador de seguridad                    3 → 0

configurados como elegibles
  con aprobación de 2 personas y caducidad de 4 h
  emergencia: sin aprobación, con alerta inmediata
```

```
primeros tres meses
  activaciones                                    186
  de emergencia                                     2
    · 1 incidente real
    · 1 porque el aprobador estaba de baja y nadie cubría
      → turno de aprobadores corregido

y la prueba de emergencia
  primera ejecución: funcionó, en 90 segundos
  segunda (trimestral): la cuenta había perdido un permiso
  al reorganizar la carpeta                       ley 27
  → corregido, y añadida comprobación automática mensual
```

La auditoría, con el campo que faltaba leer:

```
al investigar un borrado accidental de una tabla
  el registro decía: «cuenta de servicio de análisis borró
  la tabla»
  → y el equipo buscó qué carga lo había hecho

al leer el campo de identidad ORIGINAL
  una persona había suplantado esa cuenta y ejecutado el
  borrado a mano
  → 3 horas de investigación por no leer un campo

corrección
  las consultas guardadas incluyen siempre la identidad
  original
  y una alerta: «suplantación de una cuenta privilegiada»
```

El resultado:

```
claves de cuenta de servicio                  341           3
  usadas desde fuera sin explicar               1           0
recursos con cuenta de servicio por defecto    71           0
asignaciones en organización                   18           4
asignaciones en carpeta                       140          22
alcance de la cuenta de despliegue      86 proyectos   4-40
                                                     recursos
papeles privilegiados permanentes              27           0
asignaciones con condición                      0          59
condiciones que un servicio ignoraba            —           0
tiempo de investigación de una acción       3 horas     4 min
```

La lección que esta clase deja: eliminar las trescientas cuarenta y una claves costó cuatro meses y no redujo el alcance ni un punto: solo cambió cómo se obtiene el testigo. Reducir el alcance fue un trabajo distinto y posterior, y ahí las condiciones por recurso hicieron el trabajo que en otras nubes exige papeles a medida. Y dos de los nueve servicios ignoraban la condición, lo que se descubrió con una prueba negativa y habría dejado dos permisos aparentemente limitados sin estarlo.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-19-gcp-production-architecture/230-workload-identity-federation-iam-conditions-y-pam/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 gcp-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 gcp-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
Se eliminan las claves y el riesgo apenas baja	Quitar la credencial resuelve el robo, no el alcance de la identidad	Trata reducir alcance como un trabajo aparte: mide qué recursos alcanza cada identidad y baja el ámbito de las asignaciones.
Todos los recursos de un proyecto comparten una identidad muy permisiva	Usan la cuenta de servicio por defecto del proyecto	Una cuenta de servicio por carga, con sus permisos, y prohibición de usar la predeterminada.
Un permiso parece limitado por una condición y no lo está	El servicio no respeta las condiciones por recurso	Comprueba con una prueba negativa servicio por servicio; donde no se respeten, usa papeles personalizados y asignación en el recurso.
La auditoría atribuye una acción a una cuenta de servicio y no a la persona	No se lee el campo de identidad original en las suplantaciones	Incluye siempre la identidad original en las consultas y alerta sobre suplantaciones de cuentas privilegiadas.
Dar el permiso de suplantar parece inofensivo y no lo es	Equivale a conceder los permisos de la cuenta suplantada	Audita quién puede suplantar cada cuenta privilegiada como si fuera una asignación directa.
El acceso de emergencia funcionó una vez y luego no	Perdió un permiso al reorganizar la jerarquía y nadie lo comprobó	Prueba trimestral del acceso de emergencia y comprobación automática de sus permisos.

■ Seguridad, ética y costo

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

■ Preguntas de comprobación

1. ¿Cuáles son las cuatro formas de no necesitar una clave de cuenta de servicio?
2. ¿Qué permiten las condiciones que no permite el ámbito por sí solo?
3. ¿Por qué el permiso de suplantar debe auditarse como el permiso suplantado?
4. ¿Qué registros están desactivados por defecto y por qué hacen falta?

5. ¿Qué campo hay que leer para saber quién ejecutó realmente una acción?

Referencias

- Google Cloud (2025). Best practices for using service accounts. <https://cloud.google.com/iam/docs/best-practices-service-accounts>
- Google Cloud (2025). Workload identity federation. <https://cloud.google.com/iam/docs/workload-identity-federation>
- Google Cloud (2025). IAM conditions overview. <https://cloud.google.com/iam/docs/conditions-overview>
- Google Cloud (2025). Privileged Access Manager. <https://cloud.google.com/iam/docs/pam-overview>
- Google Cloud (2025). Policy Analyzer and IAM recommender. <https://cloud.google.com/policy-intelligence/docs/policy-analyzer-overview>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 19 en PDF · Recorrido de Google Cloud en PDF · Manual integral

Anterior	Índice	Siguiente
← 229 · Resource Manager, folders, Shared VPC y guardrails	Parte 19 · Programa	231 · Red global, load balancing, PSC y Cloud DNS →

Evaluación — 230 Workload Identity Federation, IAM Conditions y PAM

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 gcp-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.

231 — Red global, load balancing, PSC y Cloud DNS

Parte: 19 — Google Cloud: arquitectura de datos y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: network · Estado: EXECUTABLECORE

Propósito

Montar la red de Google Cloud, que es la que más se diferencia de las otras dos: la red es global y una sola dirección puede servir desde todas las regiones. La clase explica qué implica eso para el balanceo, cómo funciona la conectividad privada hacia servicios propios y de terceros, y la resolución de nombres con su equivalente del problema de la clase 219: si la zona no está asociada a la red, el nombre resuelve a la dirección pública.

Resultados de aprendizaje

Al finalizar podrás:

1. Aprovechar la red global sin trasladar suposiciones regionales.
2. Elegir el balanceador adecuado entre global y regional.
3. Conectar a servicios propios y de terceros de forma privada.
4. Resolver nombres privados en todas las redes que los necesitan.
5. Controlar la salida a internet, que está abierta por defecto.

Conceptos centrales

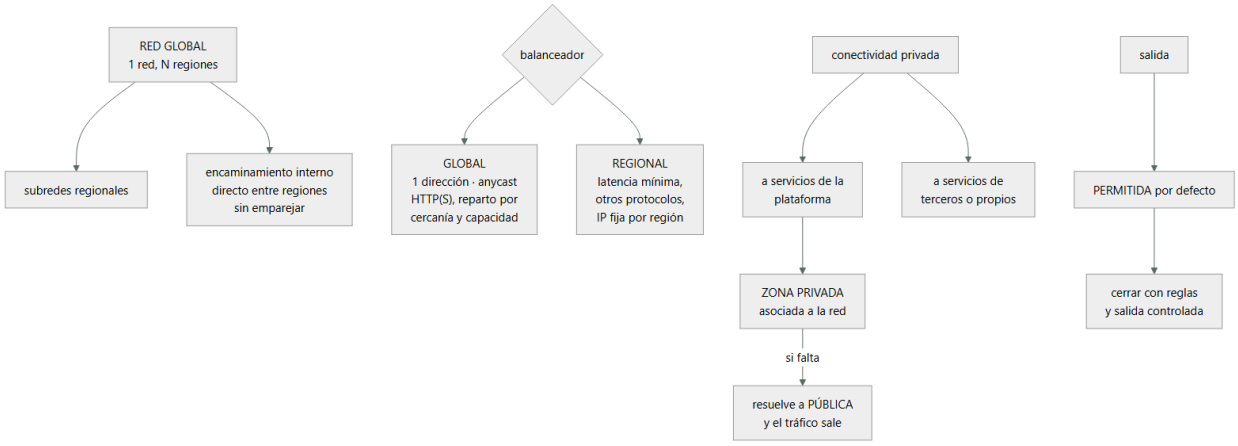
Concepto	Comprensión verificable
red global	Red virtual que abarca todas las regiones. Las subredes son regionales; el encaminamiento interno es directo.
balanceo global	Una dirección única que reparte a los grupos de la región más cercana con capacidad.
conexión de servicio privada	Mecanismo para alcanzar un servicio de la plataforma o de un tercero por una dirección de tu red.
acceso privado a servicios	Alternativa que permite alcanzar servicios de la plataforma sin dirección externa, por rutas privilegiadas.
zona privada de nombres	Zona asociada a redes concretas. Sin asociación, el nombre resuelve a la dirección pública.
salida controlada	Restricción del tráfico saliente, que está permitido por defecto y hay que cerrar.

Modelo mental

Google Cloud combina una jerarquía estricta de recursos con una red global y servicios data-first; proyecto, cuota e IAM forman una unidad operativa.

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 red global, y qué cambia

Esta es la diferencia estructural que más suposiciones invalida.

```
LA RED ES GLOBAL
  una sola red abarca todas las regiones
  las SUBREDES son regionales
  y los recursos de subredes distintas se alcanzan
  directamente, sin emparejar nada

QUÉ DESAPARECE
  emparejamientos entre regiones
  concentradores de tránsito para unir regiones propias
  y buena parte de la complejidad de encaminamiento de las
  clases 194 y 199

QUÉ NO DESAPARECE
  el coste del tráfico entre regiones, que sigue
  el coste del tráfico entre zonas, que también
  y la latencia física
```

Y las suposiciones que hay que desmontar:

```
✗ «hay que emparejar las redes de cada región»
  no; es una sola red

✗ «una subred agotada solo afecta a su región»
  cierto, pero el rango se decide en el plan global
                                                clase 193
  → y las subredes se pueden AMPLIAR sin recrear, que es
    una ventaja real frente a otras nubes

✗ «las reglas de cortafuegos son por subred»
  son de la RED, con selectores y prioridades  clase 229
  → una regla mal puesta afecta a todas las regiones

✗ «el balanceador es regional»
  hay de los dos, y el global cambia el diseño
```

Los rangos secundarios, que hacen falta para contenedores:

```
una subred puede tener rangos SECUNDARIOS
  → los pods y los servicios del clúster usan esos rangos
  → y se dimensionan aparte del rango principal
                                        clases 193, 234

→ el plan de direcciones tiene que contarlos, y son los que
  más consumen
```

Y una decisión que conviene tomar pronto:

```
¿SUBREDES AUTOMÁTICAS O PERSONALIZADAS?
  automáticas   la red crea una subred por región, con
                rangos fijos
```

```
                → cómodo y garantiza solapamiento con
                  cualquier otra red creada igual
  personalizadas se declaran los rangos del plan
                → la elección correcta para cualquier
                  organización                clase 193

→ y las redes automáticas se convierten en personalizadas,
  pero no al revés
```

2. Balanceo: global y regional

El balanceador global es la pieza que más cambia el diseño frente a las otras nubes.

```
BALANCEADOR GLOBAL DE APLICACIÓN
  UNA dirección, anunciada desde muchos puntos
  el usuario llega al punto más cercano
  y desde ahí se reparte a la región con capacidad

  qué da
    una sola entrada para todas las regiones
    conmutación entre regiones sin tocar nombres
      → y por tanto SIN el retraso del TTL ni de las cachés
        de cliente                              clase 215
    terminación TLS y certificados gestionados
    caché integrada en el borde
    y filtrado de aplicación delante

  qué implica
    la salud de los grupos por región decide el reparto
    → y hay que configurarla bien, con las lecciones de la
      clase 196

BALANCEADOR REGIONAL
  para protocolos que no son HTTP
  para latencia mínima dentro de una región
  y cuando hace falta una IP fija por región para listas
  de terceros                                   clase 193

BALANCEADOR INTERNO
  para el tráfico entre servicios propios
  regional o global según el caso
```

Y la consecuencia sobre continuidad, que es notable:

```
con balanceo global, perder una región NO exige cambiar un
registro de nombres
  → el reparto deja de enviar tráfico a esa región
  → y el tramo de «redirección» de la clase 215 casi
    desaparece

→ lo que NO desaparece
  promover los datos                        clase 235
  escalar la capacidad en la región que queda clase 233
  y decidir                                 ← sigue siendo
                                              el tramo que
                                              domina
```

Y una advertencia:

```
la capacidad de los grupos por región hay que dimensionarla
para absorber el tráfico de la región perdida
→ si cada región está al 80 %, perder una no funciona
→ es la misma aritmética de la clase 186, aplicada a
  regiones en vez de a zonas
```

Y el caché del borde:

```
el balanceador global integra caché
  → las lecciones de la clase 197 aplican igual: clave de
    caché, contenido rancio, invalidación por etiqueta
  → y el ahorro de salida y de origen también
```

3. Conectividad privada y nombres

Aquí hay dos mecanismos que se confunden y una trampa idéntica a la de la clase 219.

```
ACCESO PRIVADO A SERVICIOS
  las máquinas sin dirección externa alcanzan las API de
  la plataforma por rutas privilegiadas
  + se activa por subred, sin crear recursos
  + gratuito
  − el tráfico sale por las direcciones públicas del
    servicio, no por una dirección de tu red
  − y por tanto el control por origen es más grueso

CONEXIÓN DE SERVICIO PRIVADA
  crea una dirección DE TU RED que representa el servicio
  + alcanzable desde el centro de datos y desde otras redes
  + permite política por origen fina
  + y sirve también para servicios de TERCEROS y para
    publicar los propios a otros clientes
  − cuesta, y consume direcciones           clase 193

CRITERIO
  ¿hace falta alcanzarlo desde fuera de la red o controlar
   por origen?  → conexión privada
  ¿solo desde dentro y sin dirección externa?
               → acceso privado a servicios
```

La trampa de los nombres, que es la misma de siempre:

```
la conexión privada crea una dirección
y hace falta que el NOMBRE del servicio resuelva a ella
→ eso lo hace una ZONA PRIVADA, asociada a las redes

✗ si la zona no está asociada a una red
  el nombre resuelve a la dirección PÚBLICA
  → el tráfico sale a internet
  → y si el acceso público está cerrado, falla; y si no,
    funciona pagando salida y sin control
                                        clase 219, ley 26

→ y con red compartida, la asociación se hace en el
  proyecto anfitrión, no en el de servicio
  → es el error de configuración más frecuente aquí
```

Y la comprobación obligatoria:

```
para cada servicio con conexión privada, resolver su nombre
desde CADA red y comprobar que devuelve dirección privada
→ función de aptitud, diaria                  clase 190
```

La resolución con el centro de datos, con lo suyo:

```
políticas de reenvío entrante y saliente
  entrante   el centro de datos resuelve nombres de la nube
  saliente   la nube resuelve nombres del centro de datos
→ y hay que montar las DOS
→ el fallo clásico es montar una y descubrir la otra el día
  que un sistema corporativo tiene que llamar   clase 195
```

Y los controles de perímetro, que aquí tienen forma propia:

```
un PERÍMETRO DE SERVICIO agrupa proyectos y restringe qué
API pueden usarse desde fuera y hacia fuera
  → impide que una identidad válida copie datos a un
    proyecto de otra organización
  → es el equivalente del perímetro de datos de la
    clase 200, y es lo que cierra el hueco que deja la
    conexión privada
```

4. Salida, cortafuegos y diagnóstico

La salida está permitida por defecto, como en las otras dos nubes, y hay que cerrarla.

```
LAS REGLAS DE CORTAFUEGOS
  son de la red, con prioridad, y con reglas implícitas al
  final
    entrada: denegar todo
    SALIDA: permitir todo                      ley 26

  selectores
```

```
    por etiqueta de red   ← cualquiera con permiso sobre la
                            instancia puede ponerla
    por CUENTA DE SERVICIO ← exige permiso sobre la cuenta
  → usar cuenta de servicio siempre que se pueda
                                              clase 229

CERRAR LA SALIDA
  regla de denegación de salida con prioridad alta
  y permisos explícitos hacia lo declarado
  con las API de la plataforma alcanzadas por acceso
    privado o conexión privada

Y EL PERÍMETRO DE SERVICIO
  impide sacar datos a proyectos ajenos aunque la red lo
  permitiera                                  clase 200
```

Y los mecanismos de salida gestionada:

```
LA PASARELA DE TRADUCCIÓN DE DIRECCIONES
  da salida a máquinas sin dirección externa
  con direcciones fijas para listas de terceros
  y hay que dimensionar los puertos por instancia
  → el agotamiento de puertos aparece igual que en la
    clase 221

EL CORTAFUEGOS GESTIONADO
  inspección de capa 7 y control por nombre de destino
  → lo que hace falta para la lista de destinos permitidos
                                              clase 200
```

El diagnóstico, con las herramientas propias:

```
COMPROBADOR DE CONECTIVIDAD
  dice, para un origen y un destino, si el tráfico llegaría
  y qué regla o ruta lo permite o lo bloquea
  → responde en segundos lo que una captura tarda horas
                                              clase 202

REGISTROS DE FLUJO
  quién habla con quién, y qué se deniega
  → con muestreo configurable: al 100 % en lo crítico

ESPEJO DE PAQUETES
  copia del tráfico hacia un destino de análisis
  → caro y con datos sensibles; con procedimiento

Y LOS REGISTROS DE REGLAS DE CORTAFUEGOS
  desactivados por defecto, por regla        ley 26
  → activarlos en las reglas de denegación es lo que
    revela lo que se está bloqueando
```

Y la lista de comprobación de la clase:

- ■ la red es personalizada, con rangos del plan
- ■ los rangos secundarios están dimensionados
- ■ no se han creado redes por región innecesariamente
- ■ las reglas seleccionan por cuenta de servicio
- ■ la salida está cerrada explícitamente
- ■ hay perímetro de servicio para los proyectos con datos
- ■ el balanceador global se usa donde aporta, y el regional donde hace falta IP fija u otro protocolo
- ■ la capacidad por región absorbe la pérdida de una
- ■ las zonas privadas están asociadas a todas las redes, en el proyecto anfitrión
- ■ hay comprobación diaria de resolución a dirección privada
- ■ el reenvío con el centro de datos está en los dos sentidos
- ■ los registros de flujo están al 100 % en lo crítico
- ■ los registros de las reglas de denegación están activados

Y el cierre que enlaza con la clase siguiente: con jerarquía, identidad y red resueltas, hace falta declararlo como código con validación previa de las políticas, para que el rechazo no llegue en el momento de desplegar. Es la materia de la clase 232.

Ejemplo trabajado

CloudShop monta la red de su organización en Google Cloud. Lo que sigue es lo que ahorró la red global, el problema de zonas privadas que se repitió pese a conocerlo, y la conmutación de región que resultó ser mucho más rápida que en las otras dos nubes.

Lo que ahorró la red global:

```
el diseño trasladado de las otras nubes preveía
  3 redes (una por región)
  emparejamientos entre ellas                        3
  o un concentrador de tránsito                      1
  reglas duplicadas por región

el diseño real
  2 redes compartidas (producción y no producción)
  subredes regionales dentro de cada una
  emparejamientos                                    0
  concentradores                                     0
  reglas de cortafuegos, únicas para las 3 regiones

ahorro estimado frente al diseño trasladado
  coste de concentrador y tránsito         1.900 €/mes
  reglas que mantener                          3× menos
  y la complejidad de encaminamiento entre regiones,
  eliminada
```

Y el hallazgo que no se esperaba:

```
las subredes se pueden AMPLIAR sin recrear
  → en las otras dos nubes, ampliar exigía crear otra
    subred o renumerar                          clase 193
  → aquí, dos subredes que se estaban agotando se
    ampliaron de /22 a /20 en minutos, sin cortar nada
  → y eso cambió la estrategia de dimensionado: se puede
    empezar ajustado y ampliar
```

El problema de las zonas privadas, otra vez.

```
se montaron conexiones privadas a 19 servicios
con zonas privadas para que los nombres resolvieran a
dirección privada

y al comprobar, dos semanas después
  proyectos de servicio desde los que resolvía correcto  4
  proyectos desde los que resolvía a dirección PÚBLICA  38

causa
  con red compartida, la zona privada debe asociarse a la
  red del PROYECTO ANFITRIÓN
  el equipo la había asociado en cada proyecto de
  servicio, donde no tiene efecto
  → y funcionaba en los 4 proyectos donde alguien la había
    asociado bien por casualidad

lo incómodo
  el equipo CONOCÍA el problema: lo habían sufrido en la
  clase 219, con Azure
  → y aun así lo repitieron, porque el sitio donde hay que
    asociarla es distinto
  → es exactamente el error de traslado de la clase 229

corrección
  zonas asociadas a las redes de los proyectos anfitriones
  comprobación diaria: resolver los 19 nombres desde una
  máquina de cada proyecto de servicio
  → 4 fallos en los 6 meses siguientes, todos en proyectos
    creados antes de que la automatización existiera
                                                  ley 27
```

El balanceo global, y lo que cambió en continuidad.

```
montaje
  1 balanceador global de aplicación
```

```
  1 dirección, anunciada desde todos los puntos
  grupos de instancias en 2 regiones (Bélgica y Fráncfort)
  certificados gestionados
  filtrado y caché en el borde                clase 197

prueba de pérdida de región
  se retiraron los grupos de la región principal

  detección por comprobación de salud            28 s
  reparto a la región restante           automático
  cambio de nombre necesario                  NINGUNO
  caché de cliente                         irrelevante
  → los dos tramos que dominaban en las otras nubes
    desaparecen                               clase 215

  lo que SÍ hubo que hacer
    promover la base de datos                 2 min 10
    escalar la región restante                4 min 40
    y DECIDIR                                 3 min 20

  tiempo total                                8 min 15
  frente a 10 min 40 en Azure y 12 min 40 en AWS

y la lección
  el tramo de redirección desapareció por completo
  el tramo de DECISIÓN sigue siendo de los mayores
  → y ese no lo resuelve ninguna tecnología     clase 215
```

Y el dimensionado que hizo falta:

```
la primera prueba falló
  cada región estaba al 74 % en el pico
  al perder una, la otra no absorbía el tráfico
  → 6 minutos de degradación mientras escalaba

corrección
  cada región dimensionada para absorber el 100 % del
  tráfico total con utilización ≤ 65 %
  coste adicional                          2.100 €/mes
  → y ese es el precio real de la continuidad activa
                                        clases 186, 215
```

La salida, cerrada:

```
fase 1   registrar sin bloquear
  se activaron los registros de las reglas y los de flujo
  destinos externos observados, únicos             1.140
    necesarios y conocidos                            48
    de la propia plataforma                          620
      → deberían ir por acceso privado, no por internet
    desconocidos                                     472
      · 410 dominios de telemetría de una biblioteca
      ·  38 un servicio de traducción no declarado
      ·  14 almacenes de otras organizaciones
      ·  10 dominios registrados hacía menos de 60 días

fase 2   acceso privado a servicios activado en todas las
         subredes
  → los 620 destinos de la plataforma dejan de salir
  → salida a internet                  41 TB → 6 TB/mes

fase 3   denegación de salida por defecto, con permitidos
         por nombre a través del cortafuegos gestionado

fase 4   perímetro de servicio sobre los proyectos con
         datos
  → y aquí se comprobó lo que la clase 200 advertía:
    con la salida cerrada, aún se podía copiar a un
    almacén de otra organización usando la API de la
    plataforma
  → el perímetro lo impidió

coste de salida                     3.900 € → 610 €/mes
```

El diagnóstico, con la herramienta que ahorró horas:

```
incidente   el servicio de precios no alcanzaba la base
            desde la región secundaria

lo que se habría hecho antes
  revisar reglas a mano, capturar paquetes: horas

lo que se hizo
  comprobador de conectividad: origen la instancia,
  destino la base
  → respuesta en 12 segundos: «bloqueado por la regla
    deny-egress-default; no hay regla que permita el
    destino»
  → causa: la regla de permitido se había creado con
    selector por etiqueta y las instancias nuevas usaban
    cuenta de servicio

tiempo total                                    9 min
```

El resultado:

```
redes                                        41          2
emparejamientos                              67          0
concentradores de tránsito                    1          0
proyectos con resolución privada correcta    4/42       42/42
destinos externos alcanzables            ilimitado       48
salida a internet                          41 TB       6 TB
coste de salida                          3.900 €       610 €
tiempo de conmutación de región         sin medir    8 min 15
tiempo de diagnóstico de red              2-3 h        9 min
coste de conectividad                    6.400 €     3.900 €
```

La lección que esta clase deja: la red global eliminó los emparejamientos, los concentradores y —en la conmutación— el tramo de redirección entero, que en las otras dos nubes era el que más se alargaba. Y el error de las zonas privadas se repitió pese a que el equipo lo había sufrido en la nube anterior: el mecanismo era el mismo y el sitio donde se asocia, distinto. Es la forma más pura del error de traslado.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-19-gcp-production-architecture/231-red-global-load-balancing-psc-y-cloud-dns/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 gcp-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 gcp-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
Se crean redes por región y emparejamientos entre ellas	Se trasladó el modelo regional de otras nubes	La red es global: una red por entorno con subredes regionales, sin emparejamientos internos.
El nombre de un servicio con conexión privada resuelve a dirección pública	Con red compartida, la zona privada debe asociarse a la red del proyecto anfitrión	Asocia en el anfitrión y comprueba a diario la resolución desde una máquina de cada proyecto de servicio.
Al perder una región el servicio se degrada mientras escala	Cada región estaba dimensionada solo para su propio tráfico	Dimensiona cada región para absorber el total con utilización por debajo del codo.
Una regla de cortafuegos se puede burlar poniendo una etiqueta	El selector es por etiqueta de red	Selecciona por cuenta de servicio, que exige permiso sobre esa cuenta.
Con la salida cerrada aún se pueden copiar datos fuera	La API de la plataforma se alcanza por rutas privilegiadas y llega a proyectos ajenos	Añade un perímetro de servicio sobre los proyectos con datos.
Diagnosticar por qué no llega el tráfico lleva horas	Se revisan reglas a mano en vez de usar la comprobación de conectividad	Usa el comprobador de conectividad primero y activa los registros de las reglas de denegación.

■ 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 desaparece y qué no al ser la red global?
2. ¿Qué tramo de la conmutación de región elimina el balanceo global?
3. ¿Dónde hay que asociar una zona privada cuando se usa red compartida?
4. ¿Qué diferencia hay entre el acceso privado a servicios y la conexión de servicio privada?
5. ¿Qué cierra el perímetro de servicio que no cierra el cortafuegos?

Referencias

- Google Cloud (2025). VPC network overview. <https://cloud.google.com/vpc/docs/vpc>
- Google Cloud (2025). Cloud Load Balancing: global external Application Load Balancer. <https://cloud.google.com/load-balancing/docs/https>
- Google Cloud (2025). Private Service Connect. <https://cloud.google.com/vpc/docs/private-service-connect>
- Google Cloud (2025). Cloud DNS private zones and forwarding. <https://cloud.google.com/dns/docs/zones>

- Google Cloud (2025). Connectivity Tests. <https://cloud.google.com/network-intelligence-center/docs/connectivity-tests/concepts/overview>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 19 en PDF · Recorrido de Google Cloud en PDF · Manual integral

Anterior	Índice	Siguiente
← 230 · Workload Identity Federation, IAM Conditions y PAM	Parte 19 · Programa	232 · Terraform, Infrastructure Manager y policy validation →

Evaluación — 231 Red global, load balancing, PSC y Cloud DNS

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 gcp-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.

232 — Terraform, Infrastructure Manager y policy validation

Parte: 19 — Google Cloud: arquitectura de datos y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: iac · Estado: EXECUTABLECORE

Propósito

Declarar la infraestructura de Google Cloud como código con la pieza que faltaba en las partes anteriores: validar las políticas antes de desplegar, para que el rechazo no llegue en el momento de aplicar. La clase cubre la gestión del estado y sus riesgos, la estructura por ciclo de vida, la validación previa de restricciones, y la operación que resuelve el problema de la ley 27: qué hacer con lo que ya existe y no cumple.

Resultados de aprendizaje

Al finalizar podrás:

1. Gestionar el estado de forma segura y con bloqueo.
2. Estructurar el código por ciclo de vida y por ámbito.
3. Validar políticas y coste antes de aplicar, no después.
4. Importar lo existente al código sin recrearlo.
5. Retirar recursos de forma controlada y comprobada.

Conceptos centrales

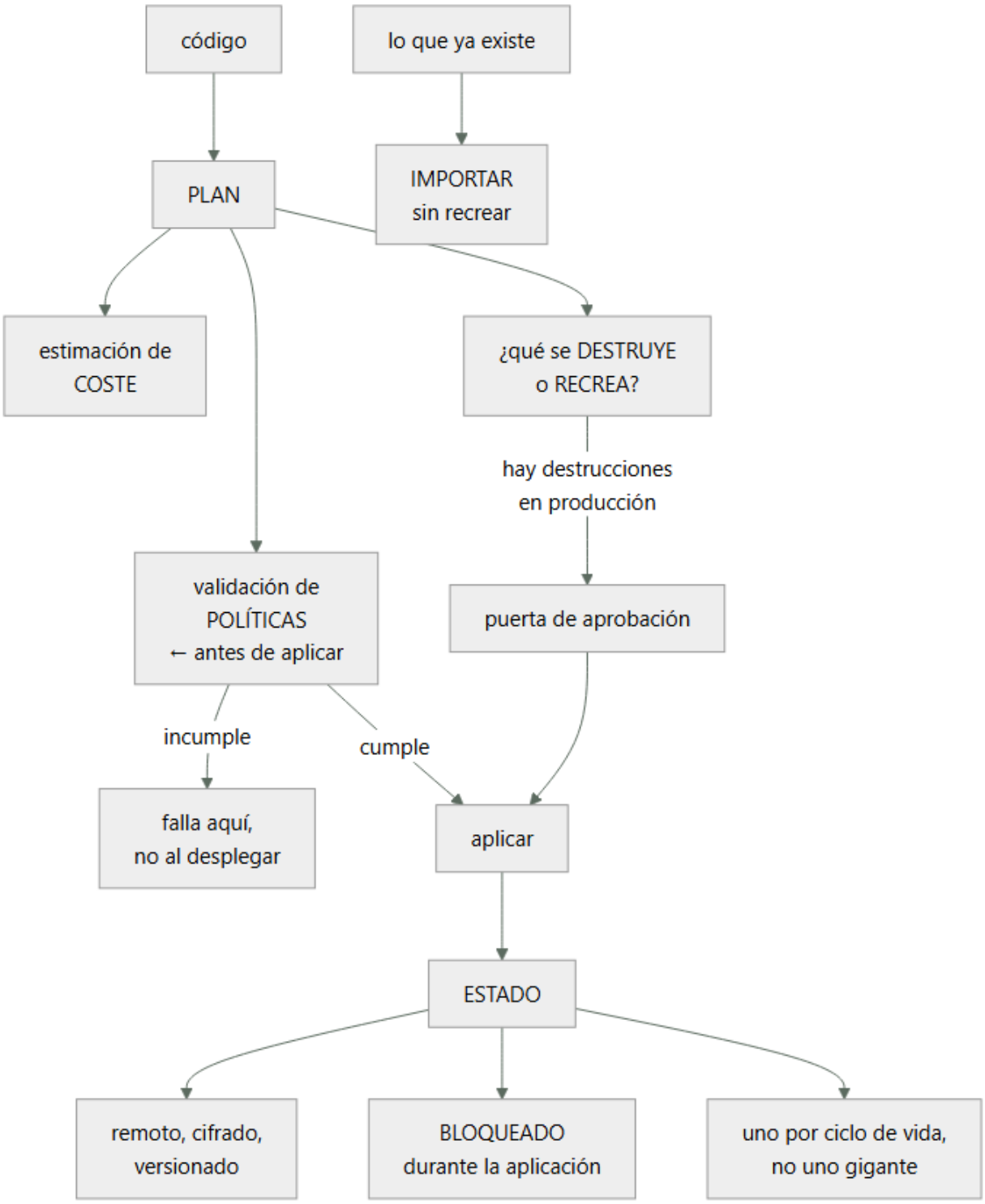
Concepto	Comprensión verificable
estado	Registro de qué recursos gestiona el código. Contiene información sensible y debe protegerse y bloquearse.
bloqueo del estado	Mecanismo que impide dos aplicaciones simultáneas. Sin él, el estado se corrompe.
plan de cambios	Previsualización de lo que se creará, modificará, recreará y destruirá. Se lee antes de aplicar.
validación de políticas	Comprobación del plan contra las restricciones de la organización, antes de aplicar.
importación	Incorporación de un recurso existente al estado, sin recrearlo.
servicio gestionado de despliegue	Alternativa que ejecuta el despliegue en la nube, con el estado gestionado por la plataforma.

Modelo mental

Google Cloud combina una jerarquía estricta de recursos con una red global y servicios data-first; proyecto, cuota e IAM forman una unidad operativa.

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 estado, y cómo no perderlo

La diferencia principal frente a Bicep es que aquí hay un estado, y eso da capacidades y riesgos.

```
QUÉ DA EL ESTADO
  saber exactamente qué gestiona el código
  detectar deriva: lo que cambió por fuera
  y destruir lo que deja de estar declarado
  → resuelve el problema de los huérfanos de la clase 220

QUÉ RIESGOS TRAE
  contiene valores sensibles en claro
```

```
    → contraseñas generadas, claves, cadenas de conexión
    → un estado en un repositorio es una fuga
  se corrompe si dos personas aplican a la vez
  y si se pierde, el código deja de saber qué gestiona
```

Y la configuración mínima:

```
ESTADO REMOTO
  en un almacén de objetos, no en el portátil
  con VERSIONADO activado
  con cifrado
  con acceso restringido: quien lee el estado ve secretos
  y con registro de acceso                    clase 238

BLOQUEO
  automático al aplicar
  → sin él, dos canalizaciones simultáneas corrompen el
    estado
  y un procedimiento para liberar un bloqueo atascado
    → que hay que tener escrito ANTES de necesitarlo

UN ESTADO POR CICLO DE VIDA
  no uno gigante
  → un estado enorme hace lentas todas las operaciones
  → y un error afecta a todo
```

Y el error que más daño hace:

```
BORRAR O PERDER EL ESTADO
  el código deja de saber qué existe
  → al aplicar, intenta CREAR lo que ya está
  → y falla con errores de recurso duplicado, o peor:
    crea duplicados donde el nombre no es único

la recuperación
  restaurar la versión anterior del estado
  → y por eso el versionado del almacén es obligatorio
  o importar todo de nuevo, que es lento y manual
```

Y la alternativa gestionada, que conviene conocer:

```
el servicio gestionado de despliegue ejecuta el código en
la nube y guarda el estado por ti
  + sin almacén que proteger, sin bloqueo que gestionar
  + con identidad de la plataforma, sin credenciales
  − menos control sobre el proceso y sobre las versiones

→ para equipos pequeños o que empiezan, suele ser mejor
→ y para canalizaciones complejas, el modelo propio
```

2. Estructura y ámbitos

La estructura es la de siempre, con los ámbitos de esta nube.

```
SEPARADO POR CICLO DE VIDA Y POR ÁMBITO
  organizacion/    políticas, carpetas, permisos de
                   organización
  red/             red compartida, subredes, cortafuegos,
                   zonas privadas
  proyectos/       creación y configuración base
  datos/           bases, conjuntos, almacenes
                   → con protección contra destrucción
  cargas/          servicios, trabajos, colas

→ cada uno con su estado
→ y las dependencias entre ellos por salidas y fuentes de
  datos, no por variables copiadas
```

Y las reglas que evitan los problemas frecuentes:

```
PROTECCIÓN CONTRA DESTRUCCIÓN
  los recursos con datos llevan la marca que impide
  destruirlos
  → y entonces un plan que los destruiría FALLA en el plan,
    no en el aplicar
  → es la protección más barata que existe
```

```
SIN SECRETOS EN VARIABLES
  referencias al gestor de secretos
  → y aun así, el valor acaba en el estado: por eso el
    estado se protege como un secreto

VERSIONES FIJADAS
  la del proveedor y las de los módulos
  → sin fijar, el mismo código produce cosas distintas en
    momentos distintos                          ley 25

MÓDULOS DEL CATÁLOGO OFICIAL
  con los valores por defecto revisados
  → misma idea que en la clase 220, contra la ley 26
```

Y los ámbitos de despliegue, que aquí importan:

```
lo que se crea en la ORGANIZACIÓN
  carpetas, políticas de organización, permisos de
  organización
  → exige permisos muy altos
  → y por eso ese estado y esa canalización van aparte,
    con aprobación                            clase 229

lo que se crea en el PROYECTO
  casi todo lo demás
  → con la identidad federada del proyecto
```

Y una advertencia de traslado:

```
✗ «una carpeta de módulos por entorno»
  produce código duplicado que diverge
✓ un módulo, y ficheros de variables por entorno
  → y la diferencia entre entornos, visible en un solo
    sitio
```

3. Validar antes de aplicar

Esta es la pieza que las partes 17 y 18 echaban de menos: comprobar que el cambio cumplirá las políticas antes de intentarlo.

```
EL PROBLEMA SIN VALIDACIÓN PREVIA
  el despliegue llega a producción y la política lo rechaza
  → el cambio queda a medias
  → y el equipo descubre la restricción en el peor momento

LA VALIDACIÓN PREVIA
  se genera el PLAN
  se convierten los recursos previstos a su forma final
  se evalúan contra las restricciones de la organización
  → y la canalización FALLA ahí, con un mensaje que dice
    qué restricción y qué falta
```

Y lo que se puede validar:

```
las restricciones de organización         clase 229
las políticas propias de la empresa
  «ningún almacén sin cifrado con clave propia»
  «ninguna base sin copia de seguridad»
  «ningún recurso sin etiquetas obligatorias»
y las convenciones
  nombres, regiones, tamaños permitidos
```

Y las otras dos puertas que deben estar en la misma canalización:

```
ESTIMACIÓN DE COSTE
  a partir del plan, cuánto costará al mes
  → publicada en la revisión                clase 220
  → y con umbral: por encima de X, aprobación

DESTRUCCIONES Y RECREACIONES
  el plan las lista
  → y una recreación destruye y crea: para una base, es
    pérdida de datos
  → puerta automática: si hay destrucciones en producción,
    aprobación explícita                    clase 220
```

Y el orden completo de la canalización:

```
1  formato y análisis estático
2  validación de sintaxis
3  PLAN, con identidad federada y solo lectura
4  validación de políticas sobre el plan
5  estimación de coste
6  publicación del resumen: qué se crea, cambia, RECREA y
   destruye
7  puerta si hay destrucciones o si el coste supera el
   umbral
8  aplicar, con identidad de despliegue
9  comprobar el resultado
```

Y una práctica que ahorra incidentes:

```
el PLAN se ejecuta con una identidad de SOLO LECTURA
→ y la de escritura solo se usa en el paso de aplicar
→ así una propuesta de cambio de cualquiera puede generar
  plan sin poder cambiar nada                clase 230
```

4. Importar, detectar deriva y retirar

La importación es lo que permite poner bajo código lo que ya existe, que es el problema de la ley 27.

```
EL ESCENARIO HABITUAL
  hay cientos de recursos creados a mano
  el código nuevo describe lo que debería haber
  → aplicarlo intentaría crear lo que ya existe

LA IMPORTACIÓN
  incorpora el recurso existente al estado
  → y a partir de ahí, el código lo gestiona
  → sin recrearlo ni cortar nada

y el trabajo real
  hay que escribir el código que COINCIDA con lo que existe
  → y el plan posterior dice qué diferencias hay
  → esas diferencias son el inventario de lo que estaba mal
```

Y el orden que funciona:

```
1  generar el código a partir de lo existente, donde la
   herramienta lo permita
2  importar
3  ejecutar el plan: si sale vacío, el código describe la
   realidad
4  y si no, cada diferencia es una decisión: ¿se ajusta el
   código o se corrige el recurso?
5  corregir hacia el estado deseado, por lotes
```

La deriva, que es lo que ocurre después:

```
alguien cambia algo por la consola durante un incidente
→ el código deja de describir la realidad
→ y el siguiente despliegue lo revierte sin avisar

LA DETECCIÓN
  ejecutar el plan periódicamente y alertar si no está
  vacío
  → un plan que no está vacío sin que nadie haya cambiado
    el código significa deriva
  → y esa alerta es la que detecta los cambios manuales
                                        ley 25, clase 219
```

La retirada, que es lo que el estado permite y casi nadie usa:

```
quitar el recurso del código y aplicar → se destruye
→ y por eso la protección contra destrucción en los datos

y para lo que hay que sacar del código sin borrarlo
  se quita del estado, y el recurso queda huérfano
  → útil cuando pasa a gestionarlo otro equipo
  → y peligroso si se olvida: queda sin gobierno   ley 20
```

Y las comprobaciones de esta clase:

- borrar el estado y comprobar que se puede restaurar
- aplicar dos veces en paralelo y ver que el bloqueo lo impide
- intentar destruir un recurso protegido
- desplegar algo que incumple una política y ver que falla en la validación, no al aplicar
- ejecutar el plan sin cambios y comprobar que está vacío
- desplegar el entorno de cero en un proyecto vacío

clase 220

Y la lista de comprobación de la clase:

- el estado es remoto, cifrado, versionado y con acceso restringido
- el bloqueo está activo y hay procedimiento para liberarlo
- hay un estado por ciclo de vida, no uno gigante
- los recursos con datos tienen protección contra destrucción
- las versiones del proveedor y de los módulos están fijadas
- se usan módulos del catálogo oficial donde existen
- la canalización valida políticas sobre el plan
- estima el coste y lo publica
- lista destrucciones y recreaciones, con puerta de aprobación
- el plan usa identidad de solo lectura
- lo existente está importado, y el plan sale vacío
- hay detección periódica de deriva, con alerta
- el entorno se despliega de cero periódicamente

Y el cierre que enlaza con la clase siguiente: con la base declarada y validada, quedan las cargas. Las opciones de cómputo gestionado de esta nube —y la que ha cambiado más el modelo de contenedores— son la materia de la clase 233.

Ejemplo trabajado

CloudShop pone su infraestructura de Google Cloud bajo código. Lo que sigue es la importación de 4.100 recursos creados a mano, el estado que se perdió una vez, y la validación previa que evitó 61 despliegues fallidos en producción.

El punto de partida:

```
recursos existentes                          9.400
  declarados en algún código                   410   4 %
  creados a mano                             8.990

estados existentes                              11
  en un almacén sin versionado                   7
  EN EL REPOSITORIO                              2  ←
  en portátiles                                  2

y los 2 del repositorio contenían
  3 contraseñas de bases de datos en claro
  1 clave de cuenta de servicio
  → visibles para los 214 miembros de la organización
  → desde hacía 14 meses                     ley 15
```

Y la corrección inmediata:

```
estados retirados del repositorio y del historial
contraseñas y clave rotadas
almacén de estados: versionado, cifrado con clave propia,
  acceso restringido a 4 identidades, registro de acceso
  activado                                clase 238
→ y una función de aptitud: ningún fichero de estado en
  ningún repositorio                      clase 190
```

La importación, por lotes.

```
orden elegido
  1  red y cortafuegos          (más estable, menos riesgo)
  2  proyectos y permisos
  3  datos                      (con protección primero)
  4  cargas                     (las que más cambian)
```

```
el proceso por lote
  generar el código desde lo existente
  importar
  ejecutar el plan
  → y aquí aparecía el trabajo real

lote de red: 210 recursos
  plan tras importar: 118 diferencias
    · 41 reglas de cortafuegos con descripción vacía
      → se ajustó el código
    · 38 subredes sin registros de flujo activados
      → se corrigió el RECURSO
    ·  9 reglas con selector por etiqueta       clase 229
      → se corrigieron
    · 30 diferencias de campos irrelevantes
      → se ajustó el código

→ cada diferencia era una decisión, y 47 de las 118
  revelaron configuraciones que estaban mal
```

Y el resultado global:

```
recursos importados                              4.100
  → los otros 4.890 eran efímeros o de servicios
    gestionados que no se declaran
diferencias encontradas al importar               1.240
  ajustes al código                                 810
  CORRECCIONES DE RECURSOS                          430  ←

→ importar fue, además, una auditoría de configuración
→ y las 430 correcciones son la remediación que la ley 27
  exige                                         clase 228
```

El estado que se perdió.

```
qué pasó
  una canalización mal configurada aplicó con la variable
  de ruta del estado vacía
  → creó un estado nuevo, vacío, en la ruta por defecto
  → y el siguiente despliegue, con el estado bueno, entró
    en conflicto

  y el estado bueno se sobrescribió

qué salvó el caso
  el almacén tenía versionado activado
  → se restauró la versión anterior en 4 minutos

qué habría pasado sin versionado
  el código no habría sabido qué existía
  → habría intentado crear 4.100 recursos que ya estaban
  → y en los que el nombre no es único, habría creado
    duplicados

correcciones
  la ruta del estado, obligatoria y comprobada en la
  canalización
  copia diaria del estado a otro almacén
  y una prueba: borrar el estado en un entorno inferior y
  restaurarlo                                     ley 22
```

La validación previa de políticas, que era lo que faltaba.

```
antes de montarla, en 3 meses
  despliegues rechazados por política EN EL MOMENTO DE
  APLICAR                                           61
    de ellos, en producción                         19
    de ellos, que dejaron el cambio a medias        11
  tiempo medio de resolución                   2 h 20

tras montarla
  la canalización convierte el plan a la forma final y lo
  evalúa contra las restricciones
  → falla en la validación, con el mensaje
```

```
despliegues rechazados en el momento de aplicar
                                          61 → 2
  los 2, por restricciones que la validación no cubre
rechazados en la validación                 74
  → antes de tocar nada, en 40 segundos
tiempo medio de resolución          2 h 20 → 9 min
```

Y las políticas propias que se añadieron a la validación:

```
ningún almacén sin cifrado con clave propia
ninguna base sin copia de seguridad configurada
ningún recurso sin las 5 etiquetas obligatorias
ninguna regla de cortafuegos con selector por etiqueta
ninguna cuenta de servicio con papel de proyecto
ningún recurso fuera de las regiones de la UE

→ y cada una con un mensaje que dice qué falta y enlaza el
  módulo que ya lo cumple                  clase 217
```

La estimación de coste, que evitó dos sorpresas:

```
caso 1   una propuesta de cambio añadía un clúster con
         nodos de 32 núcleos
         estimación                  +8.400 €/mes
         → la revisión preguntó por qué; resultó ser un
           valor copiado de otro entorno
         → corregido a nodos de 8: +1.100 €/mes

caso 2   una propuesta activaba registros de acceso a
         datos en todos los servicios
         estimación                  +3.900 €/mes
                                        clase 238
         → se acotó a los servicios con datos sensibles:
           +410 €/mes
```

La deriva, detectada:

```
plan ejecutado a diario, sin cambios de código
  → si no sale vacío, hay deriva

primeros 6 meses
  detecciones                                  23
    cambios hechos por consola durante incidentes    14
      → 9 de ellos deberían haberse revertido y no se
        revirtieron                          ley 25
    cambios de servicios gestionados que ajustan campos     6
      → se marcaron para ignorar
    3 cambios que nadie pudo explicar
      → investigados; 2 eran automatismos de la
        plataforma, 1 fue un script de un proveedor

→ y las 9 del primer grupo son exactamente lo que la deriva
  sirve para encontrar
```

Las pruebas negativas, ejecutadas:

```
✓  borrar el estado y restaurarlo            4 min
✓  aplicar dos veces en paralelo             bloqueado
✓  destruir un recurso protegido             falla en el
                                             plan
✓  desplegar algo que incumple una política  falla en
                                             validación

✗  plan sin cambios, vacío
   → 6 recursos con deriva permanente por campos que la
     plataforma ajusta sola
   → marcados para ignorar; corregido
✓  desplegar el entorno de cero en proyecto vacío  61 min
```

El resultado:

```
recursos bajo código                        4 %        97 %
estados en repositorios                       2           0
secretos expuestos en estados                 4           0
despliegues rechazados al aplicar         61/trim      2/trim
tiempo de resolución de un rechazo        2 h 20       9 min
configuraciones corregidas al importar         —         430
```

```
deriva detectada y corregida                  no      23/6mes
sorpresas de coste en despliegues           2/trim         0
tiempo de despliegue de cero            imposible      61 min
```

La lección que esta clase deja: importar cuatro mil cien recursos fue además una auditoría, y produjo cuatrocientas treinta correcciones de configuración que ninguna herramienta de postura había señalado. La validación previa de políticas convirtió sesenta y un rechazos en producción en setenta y cuatro fallos de canalización de cuarenta segundos. Y el estado estuvo catorce meses en un repositorio con tres contraseñas de producción en claro, visible para doscientos catorce personas.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-19-gcp-production-architecture/232-terraform-infrastructure-manager-y-policy-validation/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 gcp-iac-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 gcp-iac-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
El estado contiene secretos visibles para mucha gente	Está en un repositorio o en un almacén con acceso amplio	Estado remoto, cifrado, versionado y restringido a unas pocas identidades, con registro de acceso, y una función de aptitud que prohíba ficheros de estado en repositorios.
Se pierde el estado y el código intenta crear lo que ya existe	El almacén no tiene versionado o la ruta del estado no está fijada	Versionado obligatorio, copia diaria, ruta comprobada en la canalización y prueba de restauración.
Un despliegue se rechaza por política a mitad de aplicar	No se valida el plan contra las restricciones antes de aplicar	Convierte el plan a su forma final y evalúalo contra las políticas en la canalización, con mensajes que digan qué falta.
Una recreación destruye datos	El plan no se leyó y el recurso no estaba protegido	Marca los recursos con datos contra destrucción y pon puerta de aprobación cuando el plan liste destrucciones.
Un cambio hecho en la consola desaparece en el siguiente despliegue	Deriva no detectada entre el código y la realidad	Ejecuta el plan periódicamente sin cambios de código y alerta si no sale vacío.
Aparecen sorpresas de coste tras un despliegue	No se estima el coste del plan en la revisión	Añade estimación de coste a la canalización, con umbral que exija aprobación.

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 capacidades da el estado y qué riesgos trae?
2. ¿Qué ocurre si se pierde el estado y cómo se recupera?
3. ¿Qué se gana validando las políticas sobre el plan en vez de al aplicar?
4. ¿Qué revela la importación de recursos existentes además de ponerlos bajo código?
5. ¿Cómo se detecta que alguien cambió algo por la consola?

Referencias

- HashiCorp (2025). Terraform state: remote backends and locking. <https://developer.hashicorp.com/terraform/language/state>
- Google Cloud (2025). Infrastructure Manager. <https://cloud.google.com/infrastructure-manager/docs/overview>
- Google Cloud (2025). Policy validation with Policy Library. <https://cloud.google.com/docs/terraform/policy-validation>
- Google Cloud (2025). Cloud Foundation Toolkit modules. <https://cloud.google.com/foundation-toolkit>
- HashiCorp (2025). Importing existing infrastructure. <https://developer.hashicorp.com/terraform/cli/import>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 19 en PDF · Recorrido de Google Cloud en PDF · Manual integral

Anterior	Índice	Siguiente
← 231 · Red global, load balancing, PSC y Cloud DNS	Parte 19 · Programa	233 · Cloud Run, Functions, API Gateway y Workflows →

Evaluación — 232 Terraform, Infrastructure Manager y policy validation

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 gcp-iac-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.

233 — Cloud Run, Functions, API Gateway y Workflows

Parte: 19 — Google Cloud: arquitectura de datos y operación en producción Nivel: avanzado · Horas estimadas: 4 Laboratorio: serverless · Estado: EXECUTABLECORE

Propósito

Desplegar cargas sin gestionar servidores en Google Cloud, donde el modelo de contenedores gestionados es el que más ha cambiado la forma de trabajar: se despliega una imagen, escala a cero y se reparte tráfico por revisión, sin orquestador. La clase cubre la configuración que separa un servicio de demostración de uno de producción —concurrencia, instancias mínimas, tiempos de espera y conexiones—, las funciones y la orquestación de pasos largos.

Resultados de aprendizaje

Al finalizar podrás:

1. Elegir entre servicio de contenedores, funciones y flujos de trabajo.
2. Configurar concurrencia e instancias con criterio, no por defecto.
3. Conectar las cargas a la red privada, de entrada y de salida.
4. Desplegar con reparto por revisión y vuelta atrás inmediata.
5. Orquestar procesos de varios pasos sin escribir la coordinación.

Conceptos centrales

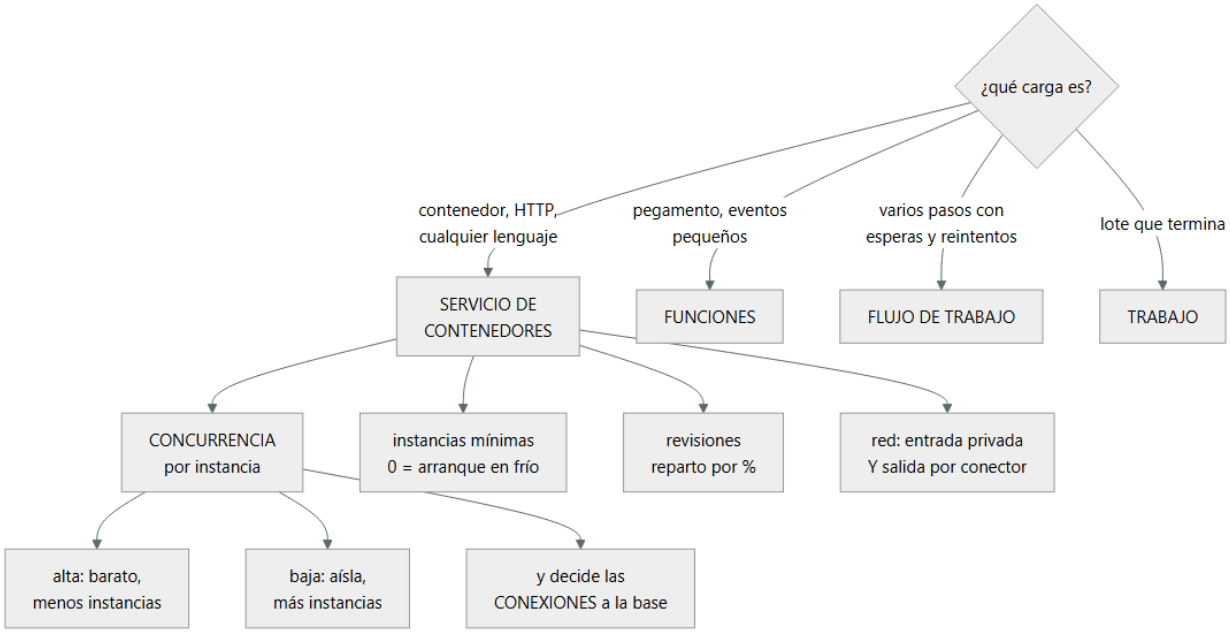
Concepto	Comprensión verificable
servicio de contenedores gestionado	Ejecuta una imagen con escalado automático, escala a cero y reparto de tráfico por revisión.
concurrencia por instancia	Cuántas peticiones atiende una instancia a la vez. Es el ajuste que más cambia coste y latencia.
instancia mínima	Número de instancias siempre encendidas. Elimina el arranque en frío y se paga.
revisión	Versión inmutable del servicio. El tráfico se reparte entre revisiones por porcentaje.
conector de red	Mecanismo que hace que el tráfico saliente del servicio pase por la red privada.
flujo de trabajo	Orquestación declarativa de pasos con reintentos, esperas y compensaciones, sin código de coordinación.

Modelo mental

Google Cloud combina una jerarquía estricta de recursos con una red global y servicios data-first; proyecto, cuota e IAM forman una unidad operativa.

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 forma de ejecutar

Hay cuatro opciones y se diferencian por lo que ejecutan y cuánto duran.

```
SERVICIO DE CONTENEDORES
  una imagen que atiende peticiones
  + cualquier lenguaje y cualquier dependencia
  + escala a cero y escala por peticiones
  + reparto de tráfico por revisión, nativo
  + sin orquestador que operar
  – arranque en frío si escala a cero

FUNCIONES
  código sin empaquetar, disparado por eventos o HTTP
  + más simple para pegamento
  – menos control del entorno
  → y en la práctica se ejecutan sobre la misma base que
    los servicios: la diferencia es de empaquetado

TRABAJO
  un contenedor que se ejecuta, termina y no atiende
  peticiones
  + para lotes, migraciones y procesos programados
  + con paralelismo por tareas

FLUJO DE TRABAJO
  orquestación declarativa de pasos
  + reintentos, esperas largas, ramas y compensaciones sin
    escribir código de coordinación
  – no es para lógica compleja: para coordinar
```

Y el criterio:

```
¿atiende peticiones HTTP?          servicio
¿reacciona a un evento y es
  pequeño?                         función
¿empieza, hace y termina?           trabajo
¿coordina varios pasos con
  esperas y reintentos?              flujo de trabajo
¿necesita operadores, GPU o
  planificación fina?                clúster       clase 234
```

Y la diferencia principal frente a lo visto en las otras nubes:

```
aquí el servicio de contenedores cubre lo que en otras
nubes se reparte entre varios productos
```

```
  → aplicaciones web, APIs, trabajadores y funciones
  → con un solo modelo de despliegue y de escalado
→ y eso reduce mucho lo que hay que aprender y operar
                                                    ley 23
```

Y una advertencia de traslado:

```
✗ «escala a cero, así que es lo más barato»
  con tráfico continuo, un servicio con instancias mínimas
  puede salir más barato que uno que arranca y para
  → y con concurrencia alta, mucho más    ← ver abajo
```

2. Concurrencia: el ajuste que más cambia

Este es el parámetro que distingue este modelo y el que más se configura mal.

```
CONCURRENCIA POR INSTANCIA
  cuántas peticiones atiende UNA instancia a la vez
  valor por defecto: alto (80 en muchos casos)

  ALTA
    + menos instancias para el mismo tráfico
    + mucho más barato
    + menos arranques en frío
    − una petición pesada afecta a las otras 79
    − y la memoria se comparte entre todas

  BAJA (1)
    + aislamiento total: como una función
    + útil si el código no es seguro para concurrencia
    − una instancia por petición: caro y con muchos
      arranques
```

Y el cálculo, que es el de la clase 186:

```
instancias = caudal × latencia / concurrencia

  1.000 pet/s × 0,08 s / 80 = 1 instancia
  1.000 pet/s × 0,08 s /  1 = 80 instancias

→ el mismo tráfico, con 80 veces más instancias
→ y por tanto, con 80 veces más conexiones a la base
```

Y de ahí la consecuencia que rompe sistemas:

```
LAS CONEXIONES A LA BASE
  cada instancia abre su agrupación
  instancias máximas × tamaño de agrupación = conexiones

  con concurrencia 1 y máximo 100 instancias
    100 × 5 = 500 conexiones
    y la base admite 200
    → agotamiento, igual que en la clase 207

  con concurrencia 80 y el mismo tráfico
    2 instancias × 5 = 10 conexiones

→ subir la concurrencia es también la forma de reducir
  conexiones
→ y donde no se pueda, hace falta un intermediario
```

Y cómo se elige el valor:

```
¿el código es seguro para atender varias peticiones a la
 vez?
  no → concurrencia 1, y arreglarlo
  sí → medir

y se mide
  con la carga real, subiendo la concurrencia hasta que la
  latencia empeora
  → el punto donde el p99 empieza a subir es el codo
                                                clase 186
  → y se elige algo por debajo
```

Las instancias mínimas, con la decisión de coste:

```
MÍNIMO 0
  + no se paga cuando no hay tráfico
  − arranque en frío en la primera petición
  → aceptable para lo asíncrono y lo interno

MÍNIMO ≥ 1
  + sin arranque en frío
  − se paga siempre
  → obligatorio en el camino crítico

Y EL TAMAÑO DE LA IMAGEN decide cuánto dura el arranque
  → imagen mínima, dependencias justas    clase 212
  → y el trabajo de inicialización, fuera de la petición
```

3. Red, despliegue y tiempos

La red, con la misma separación de la clase 221:

```
ENTRADA
  pública, interna o solo desde el balanceador
  → «interna» significa alcanzable desde la red, no desde
    internet
  → y la autorización sigue siendo necesaria: la red no
    autoriza                                clase 189

SALIDA
  por defecto, el tráfico saliente NO pasa por tu red
  → un conector o la salida directa a la red hacen que sí
  → y solo entonces se alcanza lo privado y se pasa por el
    cortafuegos                          clases 221, 231

y hay que decidir
  ¿todo el tráfico saliente por la red, o solo el privado?
  → para control de salida, todo              clase 200
```

Y el error de traslado que se repite:

```
✗ «tiene entrada interna, ya está en la red»
  la entrada y la salida se configuran por separado
  → y sin conector, el servicio no alcanza la base privada
```

El despliegue por revisiones, que aquí es nativo y muy bueno:

```
cada despliegue crea una REVISIÓN inmutable
el tráfico se reparte por porcentaje entre revisiones

  desplegar sin tráfico                    0 %
  comprobar con una etiqueta que da una URL propia
  desviar 5 %, 25 %, 100 %
  y volver atrás cambiando el porcentaje   segundos

→ es el despliegue escalonado de la clase 102, sin montar
  nada
→ y la vuelta atrás no redespliega: cambia un número
```

Y lo que hay que cuidar:

```
LAS MIGRACIONES DE ESQUEMA
  dos revisiones sirviendo a la vez significa dos versiones
  del código contra la misma base
  → expandir y contraer, siempre              clase 188

LAS SESIONES
  las instancias son efímeras y no hay afinidad garantizada
  → el estado va fuera                       clase 149
```

Los tiempos de espera y el ciclo de vida:

```
TIEMPO MÁXIMO DE PETICIÓN
  configurable, con un máximo alto
  → y el mismo error de la clase 207: poner el máximo «por
    si acaso» significa pagar peticiones colgadas
  → múltiplo pequeño del p99                   ley 26

SEÑAL DE TERMINACIÓN
  la instancia recibe aviso antes de pararse
```

```
  → hay que atenderla: terminar lo que se está haciendo
  → si no, se cortan peticiones              clase 212

TRABAJO EN SEGUNDO PLANO
  fuera de una petición, la CPU puede no estar asignada
  → un hilo que sigue trabajando tras responder puede no
    avanzar
  → o se activa la asignación continua de CPU, o el trabajo
    va a una cola                            clase 237
```

Y una advertencia sobre eso último, que sorprende:

```
el patrón «responder rápido y seguir trabajando en
segundo plano» NO funciona por defecto aquí
→ y falla de forma intermitente, que es lo peor
→ el trabajo diferido va a una cola, siempre
```

4. Flujos de trabajo y operación

Los flujos de trabajo resuelven la coordinación sin escribirla, y evitan uno de los antipatrones de la parte 09.

```
QUÉ RESUELVEN
  «llama a A; si falla, reintenta 3 veces con retroceso;
   luego espera a que llegue B; si tarda más de 2 horas,
   compensa A y avisa»
  → escrito en un fichero declarativo
  → con el estado de cada ejecución persistido

QUÉ EVITAN
  la coordinación escrita a mano en un servicio, que
  acumula estado y se convierte en un punto único
  → y las esperas largas ocupando concurrencia

Y LOS LÍMITES
  no son para lógica de negocio compleja
  ni para volumen muy alto
  → cada paso tiene coste, y con millones de ejecuciones
    suma
```

Y la disciplina que este programa exige y aquí también aplica:

```
cada paso con efecto, IDEMPOTENTE          clase 210
  → el flujo reintenta, y un reintento no debe duplicar
las compensaciones, explícitas y probadas
  → y recordando que la compensación hace invisible el
    fallo                                        ley 19
y el estado del flujo, consultable
  → «¿por qué este pedido lleva 3 horas?» debe tener
    respuesta sin abrir registros
```

Lo que hay que vigilar en estas cargas:

```
peticiones, errores y latencia por percentil  clase 238
instancias activas frente al máximo
  → tocar el máximo significa rechazo
utilización de la concurrencia
  → si está muy por debajo, se puede subir y ahorrar
tiempo de arranque y peticiones que lo pagan
conexiones a la base por instancia
peticiones que agotan el tiempo de espera
y en flujos: ejecuciones en curso, fallidas y su antigüedad
                                                 ley 13
```

Y el coste, con las palancas propias:

```
se factura por tiempo de instancia y por peticiones

las palancas, por efecto
  1  SUBIR LA CONCURRENCIA        ← la mayor, con diferencia
  2  reducir la latencia (menos tiempo de instancia)
  3  instancias mínimas solo donde hacen falta
  4  memoria y CPU ajustadas, medidas
  5  y la asignación continua de CPU solo si se usa
```

Y la lista de comprobación de la clase:

- ■ la forma de ejecutar corresponde a la carga

- la concurrencia se eligió midiendo, no por defecto
- el código es seguro para concurrencia, comprobado
- las conexiones máximas a la base están calculadas
- las instancias mínimas cubren el camino crítico
- la imagen es mínima y el arranque está medido
- la entrada está restringida y la salida pasa por la red
- el despliegue usa revisiones con reparto por porcentaje
- los cambios de esquema usan expandir y contraer
- el tiempo de espera es múltiplo pequeño del p99
- el servicio atiende la señal de terminación
- no hay trabajo en segundo plano fuera de la petición
- los pasos de los flujos son idempotentes
- hay alerta de ejecuciones de flujo antiguas o fallidas

Y el cierre que enlaza con la clase siguiente: cuando la carga exige operadores, control fino o cargas heterogéneas, aparece el clúster gestionado, que aquí tiene un modo que quita casi toda la operación. Es la materia de la clase 234.

Ejemplo trabajado

CloudShop despliega su plataforma en Google Cloud. Lo que sigue es el ajuste de concurrencia que redujo el coste un 71 % y las conexiones a la base de 500 a 12, el trabajo en segundo plano que fallaba de forma intermitente, y el flujo de trabajo que sustituyó a un coordinador escrito a mano.

El montaje inicial, trasladado de la nube anterior:

```
11 servicios, todos con
  concurrencia                              1
  motivo del equipo   «así se comporta como una función,
                       que es lo que teníamos»
  instancias mínimas                        0
  máximo                                  100
  tiempo de espera                        300 s
  memoria                                 512 Mi
  entrada                                 interna
  salida                                  sin conector
```

Y los cuatro problemas que aparecieron el primer mes:

```
1  la base agotó conexiones en el primer pico
2  el coste era 3,4 veces el estimado
3  los servicios no alcanzaban la base privada
4  el envío de correos fallaba de forma intermitente
```

Problema 1 y 2 · La concurrencia.

```
el cálculo, con la carga real del servicio de pedidos
  caudal en el pico                     900 pet/s
  latencia p50                           78 ms

  con concurrencia 1
    instancias = 900 × 0,078 / 1 = 71 instancias
    conexiones = 71 × 7 = 497            > 200 de la base

  con concurrencia 40
    instancias = 900 × 0,078 / 40 = 2 instancias
    conexiones = 2 × 7 = 14
```

Y la medición para elegir el valor:

```
se subió la concurrencia por escalones, midiendo el p99

  concurrencia   p99      instancias en el pico
       1        142 ms          71
      10        148 ms           8
      25        151 ms           3
      40        156 ms           2
      60        189 ms           2
      80        410 ms           2   ← el codo

  → el p99 empieza a subir claramente entre 60 y 80
  → se eligió 40, con margen

y antes hubo que comprobar
  ¿el código es seguro para atender varias a la vez?
```

```
  → 2 de los 11 servicios NO lo eran: usaban una variable
    global para el contexto de la petición
  → corregidos; y hasta corregirlos, quedaron en
    concurrencia 1
```

Y el efecto:

concurrencia	1	40
instancias en el pico	71	2
conexiones a la base	497	14
coste de cómputo	4.100 €	1.180 €
p99	142 ms	156 ms
agotamientos de conexiones	3/mes	0

Problema 3 · La salida sin conector.

```
síntoma   los servicios no alcanzaban la base privada
          ni el almacén con conexión privada  clase 231

causa     entrada configurada como interna
          → el equipo asumió que eso los ponía «en la red»
          → la SALIDA es un ajuste distinto

corrección
  conector de red, con subred dedicada dimensionada por el
  máximo de instancias
  y encaminamiento de CUANTO tráfico salga, por la red
  → así pasa por el cortafuegos y se registra   clase 200

y la comprobación que se añadió
  desde el servicio, resolver el nombre de la base
  → debe devolver dirección privada
  y comprobar la dirección de salida observada
  → debe ser la de la pasarela de traducción
```

Problema 4 · El trabajo en segundo plano.

```
síntoma   entre el 2 % y el 11 % de los correos de
          confirmación no se enviaban
          sin errores en los registros
          sin patrón horario

causa
  el servicio respondía al cliente y luego, en un hilo
  aparte, enviaba el correo
  → fuera de la petición, la CPU puede no estar asignada
  → el hilo se quedaba a medias y la instancia se apagaba
  → y no se registraba nada porque no llegaba a fallar

corrección
  el envío pasa a una cola; un servicio consumidor lo
  procesa                                  clase 237
  → y con idempotencia, porque hay reintentos

correos no enviados                  2-11 % → 0
```

Y la observación del equipo:

```
este fallo es el más difícil de los cuatro
  no daba error
  no tenía patrón
  y funcionaba en pruebas, donde el tráfico bajo mantenía
  las instancias vivas
→ y es una diferencia de modelo, no un error de
  configuración                             clase 229
```

El flujo de trabajo que sustituyó al coordinador.

```
el proceso de devolución tenía 7 pasos
  validar → autorizar reembolso → esperar recepción del
  paquete (hasta 14 días) → inspeccionar → reembolsar →
  actualizar inventario → notificar

la implementación anterior
  un servicio que guardaba el estado en una tabla
  un trabajo programado cada 10 min que buscaba
```

```
    devoluciones pendientes y las avanzaba
  reintentos escritos a mano
  compensaciones escritas a mano
  → 1.400 líneas de coordinación
  → y 3 incidentes en el año por estados inconsistentes

la implementación con flujo de trabajo
  fichero declarativo de 120 líneas
  reintentos con retroceso, declarados
  espera de hasta 14 días, nativa
  compensación declarada
  estado de cada ejecución, consultable

resultado
  líneas de coordinación                1.400 → 120
  incidentes por estado inconsistente       3/año → 0
  «¿por qué esta devolución lleva 3 días?»
    antes    abrir registros: 20-40 min
    después  consultar la ejecución: 30 s
  coste                                    +18 €/mes
```

Y la disciplina que hizo falta:

```
los 7 pasos, idempotentes
  → el flujo reintenta, y reembolsar dos veces cuesta
    dinero
  → clave de idempotencia por devolución e intento
                                                clase 210
y la compensación, probada
  → se ejecutó a propósito 20 veces en preproducción
  → 2 de 20 dejaron el inventario mal: corregido
                                           ley 19, 22
```

El despliegue por revisiones, y lo que evitó:

```
214 despliegues en 6 meses
  desplegados sin tráfico y comprobados con URL propia
  desviados al 5 %, luego al 100 %

  revisiones que no llegaron al 100 %             9
    · 6 detectadas en el 5 % por tasa de error
    · 3 detectadas al comprobar con la URL propia
  tiempo de vuelta atrás                        12 s
  peticiones afectadas por despliegue defectuoso
                                      ~340 (en el 5 %)

y lo que habría pasado sin reparto por porcentaje
  las 9 habrían llegado al 100 % del tráfico
```

El resultado:

```
coste de cómputo                          4.100 €     1.180 €
conexiones máximas a la base                  497          14
agotamientos de conexiones                  3/mes           0
correos no enviados                        2-11 %           0
servicios que alcanzan la base privada       0/11       11/11
líneas de coordinación de devoluciones      1.400         120
incidentes por estado inconsistente         3/año           0
despliegues defectuosos que llegaron
  a todo el tráfico                            n/d           0
```

La lección que esta clase deja: un solo parámetro —la concurrencia, puesta en 1 por trasladar el modelo de funciones— costaba dos mil novecientos euros al mes y agotaba las conexiones de la base. Subirlo a 40 redujo las instancias de setenta y una a dos. Y el fallo más difícil de los cuatro no dio ningún error: el trabajo en segundo plano fuera de la petición no avanza, y eso no es un error de configuración sino una diferencia de modelo que hay que conocer antes de trasladar código.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-19-gcp-production-architecture/233-cloud-run-functions-api-gateway-y-workflows/
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 gcp-serverless 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 gcp-serverless para el caso CloudShop. Incluye una alternativa descartada, un supuesto que pueda falsarse, una prueba de fallo y una decisión de rollback.

■ Criterio de aceptación

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

■■ Errores frecuentes

Síntoma	Causa probable	Corrección
La base agota conexiones con poco tráfico	Concurrencia por instancia baja, que multiplica el número de instancias y de agrupaciones	Mide el codo subiendo la concurrencia y elige un valor por debajo; calcula las conexiones máximas resultantes.
El coste es varias veces el estimado	Se factura tiempo de instancia y hay muchas instancias por concurrencia baja	Subir la concurrencia es la palanca de coste con más efecto; después, reducir latencia y ajustar recursos.
El servicio no alcanza recursos privados pese a tener entrada interna	Entrada y salida se configuran por separado	Añade el conector o la salida directa a la red y comprueba la dirección de salida observada.
Un trabajo diferido falla de forma intermitente y sin errores	Fuera de la petición, la CPU puede no estar asignada y el hilo no avanza	Envía el trabajo diferido a una cola con un consumidor propio, o activa la asignación continua de CPU.
Se paga tiempo de peticiones que ya no interesan a nadie	El tiempo de espera está en el máximo por si acaso	Fíjalo en un múltiplo pequeño del p99 esperado.
Dos revisiones sirviendo a la vez rompen la base	Un cambio de esquema incompatible entre versiones	Aplica expandir y contraer: el código tolera las dos formas durante la transición.

■ Seguridad, ética y costo

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

■ Preguntas de comprobación

1. ¿Cómo se relacionan concurrencia, número de instancias y conexiones a la base?
2. ¿Cómo se elige el valor de concurrencia y qué hay que comprobar antes?
3. ¿Por qué tener entrada interna no basta para alcanzar recursos privados?
4. ¿Qué le ocurre al trabajo que sigue tras responder a la petición?
5. ¿Qué resuelve un flujo de trabajo y qué sigue siendo responsabilidad del código?

Referencias

• Google Cloud (2025). Cloud Run: concurrency and instance settings. <https://cloud.google.com/run/docs/about-concurrency>

• Google Cloud (2025). Cloud Run: traffic management and revisions. <https://cloud.google.com/run/docs/managing/revisions>

• Google Cloud (2025). Cloud Run: CPU allocation and background activity. <https://cloud.google.com/run/docs/configuring/cpu-allocation>

• Google Cloud (2025). Connecting to a VPC network from Cloud Run. <https://cloud.google.com/run/docs/configuring/connecting-vpc>

• Google Cloud (2025). Workflows. <https://cloud.google.com/workflows/docs/overview>

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

Descargar: Parte 19 en PDF · Recorrido de Google Cloud en PDF · Manual integral

Anterior	Índice	Siguiente
← 232 · Terraform, Infrastructure Manager y policy validation	Parte 19 · Programa	234 · GKE Autopilot, Workload Identity y Config Sync →

Evaluación — 233 Cloud Run, Functions, API Gateway y Workflows

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 gcp-serverless 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.

234 — GKE Autopilot, Workload Identity y Config Sync

Parte: 19 — Google Cloud: arquitectura de datos y operación en producción Nivel: avanzado · Horas estimadas: 4 Laboratorio: kubernetes · Estado: EXECUTABLECORE

Propósito

Operar Kubernetes gestionado en Google Cloud, donde existe un modo que quita casi toda la operación de nodos y cambia la decisión de la clase 213: si el motivo para no usar clúster era el coste de mantenerlo, ese motivo se reduce mucho. La clase compara los dos modos con lo que cada uno permite y prohíbe, cubre la identidad de carga y el consumo de direcciones, y la reconciliación desde el repositorio con su alerta imprescindible.

Resultados de aprendizaje

Al finalizar podrás:

1. Elegir entre el modo gestionado y el estándar con criterios.
2. Decidir si el clúster compensa frente al servicio de contenedores.
3. Dar identidad a las cargas sin claves, atada al espacio de nombres.
4. Dimensionar los rangos de direcciones que consume el clúster.
5. Operar con reconciliación, políticas y actualizaciones planificadas.

Conceptos centrales

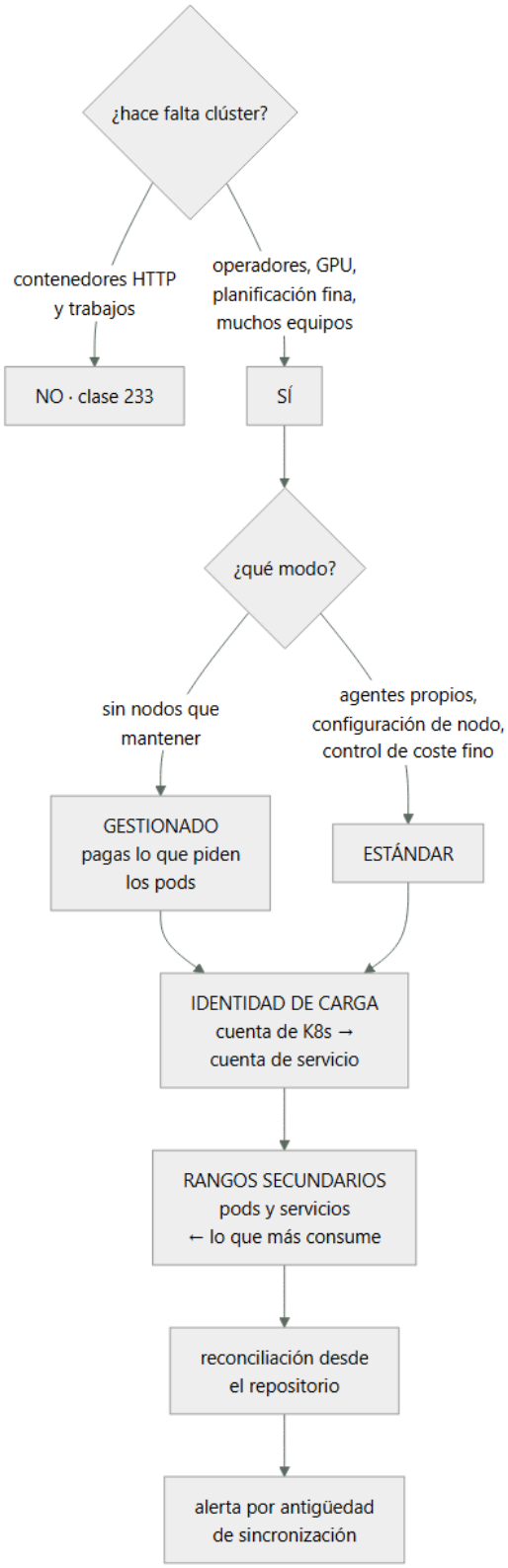
Concepto	Comprensión verificable
modo gestionado	El proveedor administra los nodos: se paga por lo que piden los pods y no hay conjunto que mantener.
modo estándar	Grupos de nodos administrados por el equipo. Más control y más trabajo.
identidad de carga	Vinculación entre una cuenta de servicio de Kubernetes y una de la nube, sin claves.
rango secundario	Rango de la subred del que salen las direcciones de pods y servicios. Es el que más consume.
canal de versión	Política de actualización automática del plano de control y de los nodos.
sincronización de configuración	Bucle gestionado que aplica el estado declarado en el repositorio y corrige la deriva.

Modelo mental

Google Cloud combina una jerarquía estricta de recursos con una red global y servicios data-first; proyecto, cuota e IAM forman una unidad operativa.

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

■ Flujo de razonamiento



Desarrollo

1. Dos modos, y qué cambia cada uno

La decisión más importante al crear el clúster es el modo, y aquí hay uno que no existe en las otras nubes.

```
MODO GESTIONADO
  no hay grupos de nodos que administrar
  se paga por CPU, memoria y disco que PIDEN los pods
```

```
  el proveedor decide y mantiene los nodos
  + sin parchear, sin escalar el conjunto, sin drenar
  + la petición de recursos es el contrato: si pides bien,
    pagas bien
  + configuración segura por defecto: política de red,
    nodos endurecidos, identidad de carga activada
                                                    ley 26
  − no se puede tocar el nodo: nada de agentes con acceso
    privilegiado ni configuraciones del sistema
  − menos control sobre el empaquetado
  − y ciertas capacidades quedan restringidas

MODO ESTÁNDAR
  grupos de nodos propios
  + control total: agentes, configuración, tipos de máquina
  + coste por unidad menor si el empaquetado es alto
  − hay que mantener el conjunto                  ley 23
  − y hay que activar a mano lo que el otro trae puesto
```

Y el efecto sobre la decisión de la clase 213:

```
el motivo principal para NO usar clúster era el coste de
operarlo
  actualizaciones, complementos, nodos, parches
→ el modo gestionado reduce mucho ese coste
→ y por tanto la frontera se mueve: hay casos que antes no
  compensaban y ahora sí

lo que NO cambia
  sigue habiendo actualizaciones de versión
  siguen los complementos que se instalen a mano
  y sigue haciendo falta saber Kubernetes
```

Y el criterio actualizado:

```
¿es un servicio HTTP o un trabajo?
  → servicio de contenedores; más simple  clase 233
¿hacen falta operadores, recursos propios o planificación
  fina?
  → clúster
¿hacen falta agentes con acceso al nodo o configuración del
  sistema?
  → clúster en modo estándar
¿ninguna de las dos?
  → clúster en modo gestionado
```

Y una advertencia de coste que sorprende:

```
en el modo gestionado se paga lo que PIDEN los pods
→ una petición de recursos generosa «por si acaso» se paga
  entera, use lo que use
→ y por tanto las peticiones bien puestas dejan de ser
  buena práctica y pasan a ser una línea de la factura
                                                clase 213
```

2. Identidad y direcciones

La identidad de carga es el mecanismo estándar aquí y conviene que esté activada desde el principio.

```
CÓMO FUNCIONA
  una cuenta de servicio de Kubernetes se anota con una
  cuenta de servicio de la nube
  el pod obtiene testigos de esa identidad
  → sin claves, sin secretos montados

Y EL ENLACE
  se concede a la cuenta de Kubernetes el permiso de
  suplantar la de la nube
  → atado a espacio de nombres Y nombre de la cuenta
  → el mismo error de las clases 218 y 222: si se ata solo
    al proyecto, cualquier cuenta de servicio del clúster
    la obtiene

Y LO QUE HAY QUE DESACTIVAR
  el acceso de los pods al servicio de metadatos del nodo
```

```
  → si no, obtienen la identidad del NODO, que suele tener
    más permisos
  → en el modo gestionado viene desactivado; en el estándar,
    hay que hacerlo                                  ley 26
```

Y la prueba obligatoria:

```
desde un pod de otro espacio de nombres, intentar obtener
la identidad
→ debe fallar
y desde un pod, leer las credenciales del nodo
→ debe estar bloqueado                              ley 22
```

Las direcciones, que es la decisión que más consume:

```
LOS TRES RANGOS
  nodos          del rango principal de la subred
  PODS           de un rango secundario
  SERVICIOS      de otro rango secundario

EL CÁLCULO
  el rango de pods se dimensiona por
    nodos máximos × pods por nodo × factor de reserva
  → y el factor no es 1: se reservan bloques por nodo
  → un clúster de 100 nodos con 110 pods por nodo puede
    necesitar un /17

Y LA TRAMPA
  el número de pods por nodo se fija al crear
  y el rango de pods NO se puede cambiar sin recrear
  → es la misma decisión irreversible de la clase 222
                                                    ley 14
```

Y las opciones que reducen el consumo:

```
reducir los pods por nodo, si no hacen falta 110
usar rangos no enrutables para pods, donde el modelo lo
  permita                                    clase 193
y reutilizar rangos secundarios entre clústeres, donde
  esté soportado

→ y en cualquier caso, calcularlo ANTES, con el plan de
  direcciones delante
```

Y el uso de red compartida:

```
el clúster vive en un proyecto de servicio y usa subredes
de la red compartida                         clase 229
→ los rangos secundarios los cede el proyecto anfitrión
→ y por tanto los pide plataforma, no el equipo
```

3. Operar desde el repositorio

La reconciliación es la misma idea de las clases 213 y 222, con una variante gestionada.

```
SINCRONIZACIÓN DE CONFIGURACIÓN GESTIONADA
  un componente del proveedor aplica lo declarado en el
  repositorio y corrige la deriva
  + menos que operar
  + con políticas de admisión integradas
  − menos control sobre versiones y extensiones

O UN CONTROLADOR PROPIO
  + control total y portable
  − hay que operarlo                             ley 23
```

Y la alerta que no puede faltar, por tercera vez:

```
«la sincronización lleva N minutos sin completarse»
o «el estado real difiere del declarado desde hace N»
→ un bucle parado no da error: deja de aplicar    ley 13
→ y en la clase 179 esa prueba negativa falló porque la
  alerta iba a un canal sin nadie
```

Las políticas de admisión, que aquí se integran con el gobierno:

```
lo que hay que impedir
```

```
  imágenes que no vengan del registro propio
  imágenes sin firmar                          clase 106
  contenedores con privilegios o con red del anfitrión
  recursos sin peticiones ni límites          clase 213
  y recursos sin etiquetas obligatorias       clase 239

y la disciplina
  modo de aviso primero, con medición
  mensajes que digan qué falta y cómo arreglarlo
  y el carril fácil que cumpla solo             ley 16
```

Y la política de red, con la diferencia entre modos:

```
en el modo gestionado viene activada
en el estándar hay que activarla, y por defecto todos los
pods hablan con todos                     ley 26, clase 222
→ denegación por defecto entre espacios de nombres, y
  reglas explícitas
```

Las actualizaciones, con lo que cambia aquí:

```
CANALES DE VERSIÓN
  rápido · regular · estable
  → el plano de control y los nodos se actualizan solos
    dentro del canal
  → con VENTANAS Y EXCLUSIONES de mantenimiento declaradas

Y ESO CAMBIA EL TRABAJO
  no hay que planificar cada actualización menor
  → pero SÍ hay que comprobar antes las interfaces
    retiradas y los complementos           clase 213
  → y las exclusiones evitan que ocurra en campaña

LO QUE SIGUE SIENDO MANUAL
  las versiones mayores
  los complementos instalados a mano
  y los operadores propios
```

Y una comprobación que hay que tener:

```
el aviso de interfaces retiradas: la plataforma detecta
manifiestos que usan API que van a desaparecer
→ conviene revisarlo antes de cada canal, no después
→ y añadirlo a la canalización              clase 232
```

4. Coste, entrada y comprobaciones

El coste, con las palancas de cada modo:

```
MODO GESTIONADO
  se paga lo pedido por los pods
  palancas
    peticiones ajustadas al p95 real         clase 213
    clases de cómputo adecuadas a la carga
    y pods de baja prioridad para lotes tolerantes
  → y no hay empaquetado que optimizar: eso lo hace el
    proveedor

MODO ESTÁNDAR
  se pagan los nodos
  palancas
    empaquetado alto
    autoescalado de nodos bien configurado
    tipos de máquina adecuados
    capacidad interrumpible para lotes
    y compromisos de uso                     clase 239

Y EN AMBOS
  el clúster tiene un coste fijo de plano de control
  → varios clústeres pequeños cuestan más que uno con
    espacios de nombres separados
  → salvo que el aislamiento lo justifique      clase 213
```

La entrada, con las opciones:

```
BALANCEADOR GLOBAL como entrada del clúster
```

```
  + una dirección para varias regiones      clase 231
  + certificados gestionados y caché en el borde
  + filtrado de aplicación delante

CONTROLADOR DE ENTRADA PROPIO
  + control y portabilidad                  clase 158
  – hay que operarlo

Y LA PUERTA DE ENLACE declarativa
  el modelo más nuevo, con separación entre quien opera la
  entrada y quien declara las rutas
  → encaja con la separación de la clase 229
```

Y las comprobaciones de esta clase:

- obtener la identidad desde otro espacio de nombres
  → debe fallar
- leer credenciales del nodo desde un pod
  → debe estar bloqueado
- desplegar una imagen de un registro ajeno
  → debe rechazarse
- desplegar un pod sin peticiones de recursos
  → debe rechazarse
- desplegar un pod con privilegios
  → debe rechazarse
- hablar entre dos espacios de nombres no permitidos
  → debe fallar
- parar la sincronización y esperar la alerta
- perder una zona y comprobar el reparto de réplicas
  clase 227

Y la lista de comprobación de la clase:

- la decisión de usar clúster está justificada frente al servicio de contenedores
- el modo elegido corresponde a lo que hace falta
- la identidad de carga está activada y atada a espacio de nombres y cuenta
- el acceso al servicio de metadatos está bloqueado
- los rangos secundarios se calcularon antes de crear
- los pods por nodo están ajustados a lo necesario
- el clúster usa la red compartida
- la política de red está activada, con denegación por defecto
- hay políticas de admisión con mensajes útiles
- la reconciliación está montada, con alerta por antigüedad
- el canal de versión y las ventanas están decididos
- se revisan las interfaces retiradas antes de actualizar
- las peticiones de recursos están medidas, no copiadas
- las réplicas se reparten entre zonas

Y el cierre que enlaza con la clase siguiente: con el cómputo resuelto, quedan los datos, que en esta nube ofrecen una familia amplia y una base distribuida con una propiedad que no tiene equivalente. Es la materia de la clase 235.

Ejemplo trabajado

CloudShop revisa su uso de Kubernetes en Google Cloud. Lo que sigue es la decisión de modo que cambió respecto a la de la clase 222, el cálculo de direcciones que evitó recrear el clúster, y el ahorro de ajustar las peticiones de recursos.

La decisión de modo:

```
lo que se necesitaba
  2 cargas con GPU del equipo de datos
  un operador de base vectorial
  trabajos por lotes con prioridades
  el buscador, con imagen propia

lo que NO se necesitaba
  agentes con acceso privilegiado al nodo
  configuración del sistema operativo
  tipos de máquina exóticos
```

→ modo GESTIONADO para todo salvo las cargas con GPU

```
→ y un grupo de nodos estándar aparte para esas
  (el modo gestionado las soporta con restricciones que no
   encajaban con el operador que usaban)

y la comparación de esfuerzo, frente al clúster de Azure
  actualizaciones al año            3 planificadas → 1
  mantenimiento de complementos     18 días/año → 4
  soporte a equipos                 44 días/año → 38
  incidentes de plataforma          11 días/año → 3
  ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  total                             82 días/año → 46
                                    0,4 → 0,22 personas
```

Y la observación que el equipo registró:

```
el modo gestionado reduce a la mitad el coste de operación
→ y eso mueve la frontera de la clase 213: hay dos cargas
  que se habían quedado fuera del clúster por el coste de
  operarlo y que ahora entrarían
→ decisión: no moverlas de todos modos; funcionan donde
  están                                          clase 216
```

El cálculo de direcciones, hecho antes de crear.

```
el equipo iba a aceptar los valores por defecto
  pods por nodo                                    110
  rango de pods sugerido                          /14

el cálculo real
  nodos máximos previstos                           38
  pods por nodo REALES observados en el clúster de
    Azure                                        máx. 31
  → 110 era el valor por defecto y no correspondía a nada

  con 32 pods por nodo
    rango de pods necesario                       /19
    rango de servicios                            /22

  frente a
  con 110 pods por nodo
    rango de pods                                 /17

→ y el plan de direcciones asignaba /18 por proyecto
→ con el valor por defecto, el clúster se habría comido
  todo el proyecto y habría bloqueado el crecimiento

y la decisión irreversible
  el número de pods por nodo se fija al crear
  → cambiarlo exige recrear el clúster           ley 14
```

La identidad, y la prueba que falló.

```
identidad de carga activada
7 cuentas de servicio de la nube, una por carga
enlace atado a espacio de nombres y nombre de cuenta

pruebas negativas
  ✓  pod de «datos» pidiendo la identidad de «buscador»
                                                 denegado
  ✓  pod leyendo credenciales del nodo           bloqueado
  ✗  el operador de base vectorial
     → su enlace se había creado atado solo al proyecto
     → cualquier cuenta de servicio del clúster podía
       obtener esa identidad
     → y esa identidad tenía permisos de escritura en el
       conjunto de datos vectoriales
     → corregido

→ y es el mismo fallo que en la clase 222, con el mismo
  componente: el que se instala a mano y no pasa por la
  plantilla                                clase 229, ley 27
```

Las peticiones de recursos, y el ahorro.

```
en el modo gestionado se paga lo PEDIDO
```

```
las peticiones estaban copiadas del clúster anterior
  carga             pedido        uso p95 real
  buscador          2 CPU / 4 Gi   0,7 CPU / 2,1 Gi
  indexador         4 CPU / 8 Gi   2,9 CPU / 5,8 Gi
  operador vect.    1 CPU / 2 Gi   0,1 CPU / 0,6 Gi
  api interna       1 CPU / 2 Gi   0,2 CPU / 0,9 Gi
  trabajos lote     2 CPU / 4 Gi   1,8 CPU / 3,6 Gi

ajuste a p95 con margen del 25 %
  buscador          0,9 CPU / 2,7 Gi
  indexador         3,6 CPU / 7,3 Gi
  operador vect.    0,2 CPU / 0,8 Gi
  api interna       0,3 CPU / 1,2 Gi
  trabajos lote     2,3 CPU / 4,5 Gi

coste del clúster gestionado      3.400 € → 1.640 €/mes

→ y aquí el ajuste de peticiones no es higiene: es
  directamente la factura                       clase 213
```

Y un efecto que no se buscaba:

```
al bajar las peticiones, más pods caben por nodo
→ y el proveedor usa menos nodos
→ pero como se paga por lo pedido, el ahorro es directo y
  no depende del empaquetado
```

La reconciliación y las políticas:

```
sincronización gestionada desde el repositorio
  con alerta de antigüedad, a canal con guardia
  → y probada parándola: alerta en 4 minutos    ley 22

políticas de admisión, en modo aviso 3 semanas
  incumplimientos medidos
    imágenes de registros ajenos                    41
      → 38 eran imágenes públicas de herramientas
      → se copiaron al registro propio y se firmaron
    pods sin peticiones de recursos                 62
      → todos, de un equipo que desplegaba a mano
    contenedores con privilegios                     4
      → 3 legítimos (agentes de red), 1 innecesario
    sin etiquetas obligatorias                      88

tras corregir, modo bloqueo
  rechazos en el primer mes                          6
  todos, incumplimientos reales
  tiquetes de soporte                                0
  → porque el mensaje decía qué faltaba      clase 217
```

Las actualizaciones, con el canal:

```
canal          regular
ventanas       martes y jueves, 02:00-06:00
exclusiones    del 15 de noviembre al 5 de enero (campaña)
               y las 2 semanas de cierre trimestral

en 12 meses
  actualizaciones automáticas aplicadas             9
  que causaron incidencia                           0
  actualizaciones mayores planificadas              1
    con revisión previa de interfaces retiradas
    → 11 manifiestos corregidos antes
  duración de la mayor                            2 h
```

El resultado:

```
coste del clúster                        3.400 €     1.640 €
direcciones consumidas por pods           8.192       2.048
  (con el valor por defecto)
capacidad de equipo dedicada          0,4 pers.   0,22 pers.
actualizaciones planificadas al año           3           1
enlaces de identidad mal atados               1           0
pods sin peticiones de recursos              62           0
imágenes de registros ajenos                 41           0
```

La lección que esta clase deja: el modo gestionado redujo a la mitad el coste de operar el clúster, lo que mueve la frontera de cuándo compensa usarlo. El valor por defecto de ciento diez pods por nodo no correspondía a nada —el uso real era treinta y uno— y aceptarlo habría consumido el proyecto entero de direcciones de forma irreversible. Y el único fallo de identidad estaba, otra vez, en el componente instalado a mano: el que no pasa por la plantilla.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-19-gcp-production-architecture/234-gke-autopilot-workload-identity-y-config-sync/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 gcp-gke-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 gcp-gke-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 el rango de direcciones del proyecto entero	Se aceptó el valor por defecto de pods por nodo sin medir el uso real	Calcula nodos por pods reales antes de crear; el valor se fija al crear y cambiarlo exige recrear el clúster.
La factura del clúster gestionado es alta pese a poco uso	Se paga lo que piden los pods y las peticiones estaban copiadas	Ajusta las peticiones al p95 real con margen; aquí es directamente la factura, no higiene.
Cualquier cuenta de servicio del clúster obtiene una identidad privilegiada	El enlace de identidad se ató solo al proyecto	Ata a espacio de nombres y nombre de cuenta, y compruébalo desde un pod de otro espacio.
Los pods obtienen permisos que no les corresponden	Pueden alcanzar el servicio de metadatos del nodo	Bloquéalo; en modo gestionado viene bloqueado, en estándar hay que hacerlo.
Los cambios del repositorio dejan de aplicarse sin aviso	El bucle de sincronización se detuvo en silencio	Alerta por antigüedad de la sincronización, con destino a guardia, y pruébala parándola.
Una actualización automática ocurre en plena campaña	No hay exclusiones de mantenimiento declaradas	Declara ventanas y exclusiones para los periodos críticos, y revisa las interfaces retiradas antes de cambiar de canal.

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 permite y qué prohíbe el modo gestionado, y cómo cambia la decisión de usar clúster?
2. ¿Por qué las peticiones de recursos son directamente la factura en modo gestionado?
3. ¿Qué rangos consume un clúster y cuál es la decisión irreversible?
4. ¿A qué debe atarse el enlace de identidad de carga?
5. ¿Qué sigue siendo manual pese a tener canal de versión?

Referencias

- Google Cloud (2025). GKE Autopilot overview. <https://cloud.google.com/kubernetes-engine/docs/concepts/autopilot-overview>
- Google Cloud (2025). Workload Identity Federation for GKE. <https://cloud.google.com/kubernetes-engine/docs/concepts/workload-identity>
- Google Cloud (2025). IP address planning for GKE clusters. <https://cloud.google.com/kubernetes-engine/docs/concepts/alias-ips>
- Google Cloud (2025). Config Sync and Policy Controller. <https://cloud.google.com/kubernetes-engine/enterprise/config-sync/docs/overview>
- Google Cloud (2025). GKE release channels and maintenance windows. <https://cloud.google.com/kubernetes-engine/docs/concepts/release-channels>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 19 en PDF · Recorrido de Google Cloud en PDF · Manual integral

Evaluación — 234 GKE Autopilot, Workload Identity y Config Sync

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 gcp-gke-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.

235 — Cloud SQL, Spanner, Firestore y Bigtable

Parte: 19 — Google Cloud: arquitectura de datos y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: data · Estado: EXECUTABLECORE

Propósito

Elegir el almacén de datos en Google Cloud, donde hay cuatro familias con propósitos claramente distintos y una que no tiene equivalente en las otras nubes: una base relacional que escala horizontalmente y da consistencia fuerte global, a cambio de un coste y unas reglas de modelado propias. La clase da el criterio de elección, las decisiones que no se pueden cambiar en cada una, y la advertencia de siempre sobre la clave.

Resultados de aprendizaje

Al finalizar podrás:

1. Elegir entre las cuatro familias con criterios comprobables.
2. Modelar para la base distribuida evitando los puntos calientes.
3. Decidir si la consistencia fuerte global compensa su coste.
4. Dimensionar capacidad y detectar el estrangulamiento.
5. Combinar el camino operativo con el analítico sin forzar ninguno.

Conceptos centrales

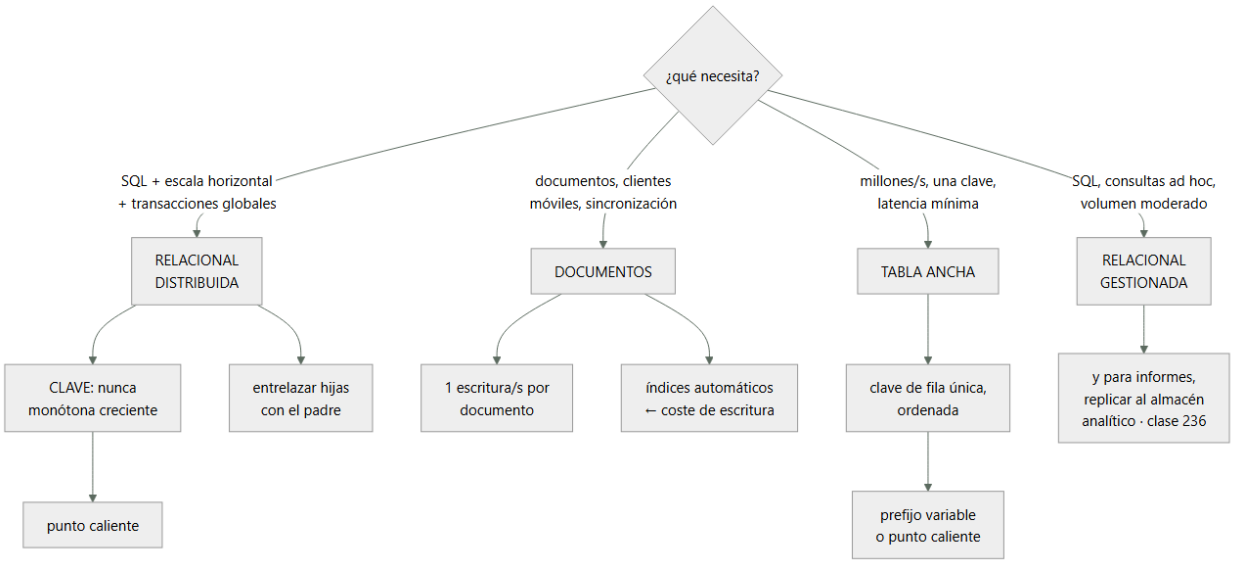
Concepto	Comprensión verificable
base relacional gestionada	Motor tradicional gestionado. Escala vertical, transacciones y consultas ad hoc.
base relacional distribuida	Escala horizontal con transacciones y consistencia fuerte global. Con reglas de modelado propias.
base de documentos	Almacén sin esquema con sincronización a clientes. Cómoda y con límites de escritura por documento.
base de tabla ancha	Almacén de altísimo volumen y baja latencia, con una sola clave ordenada.
punto caliente	Clave que concentra escrituras contiguas. En estas familias, la causa principal de mala escala.
entrelazado	Colocación física de filas hijas junto a su padre, para que la transacción no cruce servidores.

Modelo mental

Google Cloud combina una jerarquía estricta de recursos con una red global y servicios data-first; proyecto, cuota e IAM forman una unidad operativa.

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

■ Flujo de razonamiento



Desarrollo

1. Cuatro familias, cuatro propósitos

Aquí las familias están más diferenciadas que en las otras nubes, y elegir mal se paga pronto.

```
RELACIONAL GESTIONADA
  motor tradicional, gestionado
  + SQL completo, transacciones, consultas ad hoc
  + migración desde lo heredado casi directa
  – escala vertical con techo
  – y las conexiones son un recurso limitado  clase 233
  para   la mayoría de los sistemas de tamaño normal

RELACIONAL DISTRIBUIDA
  escala horizontal CON transacciones y consistencia fuerte
  global
  + no hay que elegir entre SQL y escala
  + disponibilidad muy alta y multirregión real
  – coste base alto: hay un mínimo de capacidad
  – reglas de modelado propias, y la clave manda
  para   volúmenes que una relacional no aguanta, o cuando
         se necesita consistencia fuerte entre regiones

DOCUMENTOS
  almacén sin esquema, con sincronización a clientes
  + cómodo, con consultas por campo y tiempo real
  + integración directa con aplicaciones móviles
  – límite de escrituras por documento
  – índices automáticos que se pagan en cada escritura
  para   perfiles, catálogos, estado de aplicaciones
         cliente

TABLA ANCHA
  altísimo volumen, latencia de milisegundos, una sola
  clave ordenada
  + escala a petabytes con latencia estable
  – sin consultas por otros campos: solo por la clave
  – coste base alto (nodos)
  para   series temporales, telemetría, perfiles de
         usuario a gran escala
```

Y el criterio, en preguntas:

```
¿el volumen cabe en una máquina grande y crecerá poco?
  → relacional gestionada; y no compliques

¿hace falta SQL y no cabe, o hace falta consistencia
 fuerte entre regiones?
  → relacional distribuida, midiendo el coste base
```

```
¿es estado de aplicación cliente con sincronización?
  → documentos

¿son millones de escrituras por segundo con una clave
 conocida?
  → tabla ancha

¿son informes y exploración?
  → ninguna de las cuatro: almacén analítico  clase 236
```

Y la advertencia de coste que decide muchas elecciones:

```
la relacional distribuida y la de tabla ancha tienen COSTE
BASE ALTO
  hay un mínimo de capacidad que se paga aunque no se use
→ para volúmenes pequeños salen mucho más caras
→ y por eso «empezar por la que escala» suele ser un error
  de coste                                    clase 216
```

2. La clave, otra vez

En las tres familias distribuidas, el error de diseño es el mismo y tiene el mismo nombre: la clave que concentra.

```
EL PROBLEMA
  estas bases reparten los datos por RANGOS de clave
  → claves contiguas van al mismo servidor
  → y una clave que crece de forma monótona hace que TODAS
    las escrituras nuevas caigan en el mismo sitio

LAS CLAVES QUE PRODUCEN PUNTOS CALIENTES
  identificador autoincremental
  marca de tiempo como primer componente
  y cualquier secuencia ordenada

→ y es distinto del problema de la clase 208: allí la clave
  se distribuía por resumen y el problema era la
  concentración de TRÁFICO; aquí el problema es el ORDEN
```

Y las soluciones:

```
IDENTIFICADOR ALEATORIO
  el más simple y el que mejor reparte
  – se pierde el orden natural de inserción

INVERTIR LOS BITS de un valor creciente
  reparte y conserva unicidad

PREFIJO CALCULADO
  un valor derivado del resto de la clave, como primer
  componente
  → reparte, y sigue permitiendo consultas por el resto si
    se recorre cada prefijo

CLAVE COMPUESTA con algo que reparta primero
  cliente + fecha, en vez de fecha + cliente
  → y sirve para las consultas por cliente        clase 208
```

El entrelazado, que es propio de la relacional distribuida:

```
las filas hijas se colocan FÍSICAMENTE junto a su padre
  pedido y sus líneas, juntos
→ leer un pedido con sus líneas es una sola lectura local
→ y la transacción no cruza servidores

y sin entrelazar
  la transacción coordina entre servidores
  → latencia mucho mayor y más conflictos

→ es la decisión de modelado que más cambia el rendimiento
→ y hay que tomarla al crear la tabla
```

Y las transacciones, con lo que hay que saber:

```
la relacional distribuida da transacciones globales con
consistencia fuerte
  → y eso cuesta coordinación
```

```
las de SOLO LECTURA no bloquean y son mucho más baratas
  → declararlas como tales, siempre

y las de lectura con obsolescencia acotada
  «lee de hace 10 segundos»
  → sirven la mayoría de las consultas y evitan la
    coordinación                              clase 223

→ y la mayoría de los sistemas que usan transacciones
  fuertes para todo están pagando coordinación que no
  necesitan                                      ley 26
```

3. Capacidad, límites y coste

La relacional distribuida se dimensiona por unidades de proceso.

```
LA CAPACIDAD
  se mide en unidades; hay un mínimo por instancia
  y cada unidad da un tope de almacenamiento y de caudal

  → y la regla operativa
    mantener la utilización de CPU por debajo del 65 % en
    una región y del 45 % en multirregión
    → por encima, la latencia se dispara     clase 186

EL ESTRANGULAMIENTO
  se manifiesta como latencia alta y transacciones
  abortadas
  → y las abortadas hay que reintentar, lo que empeora
  → la señal directa es la CPU y la de operaciones por
    servidor

Y LAS ESTADÍSTICAS DE ACCESO
  la plataforma indica qué rangos de clave concentran
  → es la herramienta que encuentra el punto caliente
  → y hay que mirarla, no suponer               ley 15
```

La base de documentos, con sus límites característicos:

```
UNA ESCRITURA POR SEGUNDO Y DOCUMENTO
  → un contador global en un documento no funciona
  → hay que repartirlo en varios y sumar al leer

ÍNDICES AUTOMÁTICOS
  por defecto se indexa cada campo
  → cada escritura paga por todos            clase 223
  → y hay que EXCLUIR lo que no se consulta

CONSULTAS LIMITADAS
  no hay uniones ni agregaciones complejas
  → y las consultas compuestas exigen índices declarados

Y LAS REGLAS DE SEGURIDAD
  si los clientes acceden directamente, las reglas son el
  control de acceso
  → y una regla mal escrita expone todo
  → se prueban con el simulador, siempre        ley 22
```

La tabla ancha, con lo suyo:

```
UNA SOLA CLAVE, ORDENADA
  las consultas son por clave o por rango de claves
  → no hay índices secundarios
  → el diseño de la clave ES el diseño

GRUPOS DE COLUMNAS
  separar lo que se lee junto
  → leer un grupo no trae los demás

CADUCIDAD POR CELDA
  las versiones antiguas se retiran solas
  → imprescindible en series temporales

Y EL COSTE
  por nodo y por almacenamiento
```

```
  → con un mínimo por instancia
```

Y la decisión de continuidad, que aquí es sencilla y cara:

```
la relacional distribuida en configuración multirregión da
consistencia fuerte y disponibilidad muy alta
  → y cuesta del orden del triple

y la relacional gestionada
  réplicas de lectura, y conmutación con su plazo medido
                                              clase 215
```

4. Combinar operativo y analítico

El patrón que este programa repite desde la clase 150 aquí es especialmente directo.

```
LO OPERATIVO en la familia que corresponda
LO ANALÍTICO en el almacén de análisis      clase 236

y la conexión
  flujo de cambios hacia el almacén analítico
  o consulta federada desde el almacén hacia la base
  o réplica gestionada, donde exista

→ y ninguna de las cuatro familias se fuerza para hacer
  informes
```

Y las tres cosas que hay que decidir al montarlo:

```
¿CUÁNTO RETRASO SE ACEPTA?
  segundos, minutos u horas
  → y hay que declararlo y vigilarlo            ley 13

¿QUÉ DATOS NO CRUZAN?
  los personales completos, seudonimizados o excluidos
                                        clases 230, 239

¿QUIÉN ES EL ESCRITOR?
  el almacén analítico es de solo lectura desde el punto de
  vista del negocio
  → si alguien escribe ahí y luego vuelve al operativo, hay
    dos escritores                               ley 21
```

El caché, que en muchos sistemas es lo que evita cambiar de familia:

```
un caché en memoria delante de la relacional gestionada
resuelve la mayoría de los problemas de lectura
  → y es mucho más barato que migrar a una distribuida
  → con las cautelas de la clase 111: el caché portante y
    la avalancha

→ antes de cambiar de familia por rendimiento de lectura,
  medir qué resuelve un caché
```

Y las comprobaciones de esta clase:

- escribir 10.000 filas con clave secuencial y observar el reparto
- consultar las estadísticas de acceso y comprobar que no hay rango caliente
- ejecutar una transacción que cruce servidores y medir
- escribir el mismo documento 10 veces por segundo
- probar las reglas de acceso con el simulador
- conmutar la base y cronometrar clase 215
- y comprobar el retraso del flujo hacia lo analítico

Y la lista de comprobación de la clase:

- la familia elegida corresponde a los patrones y al volumen
- se comparó el coste base antes de elegir una que escala
- la clave no es monótona creciente
- las tablas hijas están entrelazadas donde procede
- las lecturas usan transacciones de solo lectura
- se usa obsolescencia acotada donde basta
- la utilización se mantiene por debajo del umbral
- se revisan las estadísticas de acceso por rango

- en documentos, los índices automáticos están recortados
- ningún documento recibe más de una escritura por segundo
- las reglas de acceso se han probado con el simulador
- lo analítico va a su almacén, con retraso declarado
- el almacén analítico no es escritor de nada operativo

Y el cierre que enlaza con la clase siguiente: con los datos operativos resueltos, queda la plataforma analítica, que en esta nube es la pieza más madura y la que más problemas de gobierno y de coste produce. Es la materia de la clase 236.

Ejemplo trabajado

CloudShop elige sus almacenes en Google Cloud. Lo que sigue es la decisión que evitó pagar una base distribuida sin necesitarla, el punto caliente que apareció igualmente en otra tabla, y el contador de un documento que no escalaba.

La elección, carga por carga:

```
carga            volumen            elección
pedidos          41 M filas,        relacional gestionada
                 900 escrituras/s
catálogo         4,1 M productos    relacional gestionada
sesiones         2 M activas,       documentos
                 sincronización
                 con la app
telemetría de    41.000 eventos/s   tabla ancha
  la web
inventario       120 M filas,       RELACIONAL DISTRIBUIDA
  global         consistencia
                 fuerte entre 2
                 regiones
informes         —                  almacén analítico
                                    clase 236
```

La decisión que se evitó:

```
la propuesta inicial ponía CUANTO existe en la relacional
distribuida
motivo   «escala, y así no hay que migrar nunca»

el cálculo de coste
  mínimo de capacidad por instancia
  pedidos                        ~1.900 €/mes
  catálogo                       ~1.900 €/mes
  inventario                     ~2.400 €/mes
  ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  total                          ~6.200 €/mes

frente a
  pedidos y catálogo en relacional gestionada
                                    920 €/mes
  inventario en distribuida       2.400 €/mes
  ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  total                           3.320 €/mes

→ y pedidos con 41 M de filas y 900 escrituras/s cabe
  holgadamente en una relacional gestionada
→ ahorro                           2.880 €/mes
```

Y el criterio que se registró:

```
«se usa la relacional distribuida solo donde se necesita
 consistencia fuerte entre regiones o el volumen no cabe»
qué la reabriría   si pedidos supera las 4.000
                   escrituras/s sostenidas o los 500 M de
                   filas                        clase 190
```

El punto caliente del inventario.

```
el modelo inicial
  tabla inventario, clave primaria: identificador
  autoincremental
  motivo   «como en la base relacional de siempre»

lo que pasó en la prueba de carga
```

```
  a partir de 1.200 escrituras/s
    latencia p99                    de 12 ms a 890 ms
    transacciones abortadas         14 %
    CPU media de la instancia       31 %   ← engañosa

  las estadísticas de acceso por rango decían
    el 96 % de las escrituras caían en el ÚLTIMO rango
    → todas las claves nuevas eran contiguas
    → un solo servidor las atendía

corrección
  clave primaria: identificador aleatorio
  y para las consultas por producto y almacén, clave
  compuesta (producto, almacén) que reparte por producto

resultado
  latencia p99 a 1.200 escrituras/s     890 ms → 14 ms
  transacciones abortadas                  14 % → 0,2 %
  caudal máximo probado                 1.200 → 9.400/s
```

Y el entrelazado, que faltaba:

```
las tablas de movimientos de inventario no estaban
entrelazadas con la de inventario
  → cada transacción de ajuste coordinaba entre servidores
  → latencia de la transacción            48 ms

con movimientos entrelazados bajo inventario
  → la transacción es local
  → latencia                               9 ms

→ y esta decisión se toma AL CREAR la tabla; cambiarla
  exige recrear y migrar               ley 14
```

El contador del documento.

```
la aplicación móvil mantenía un contador de artículos en
el carrito, en un documento por usuario
  → funcionaba

y para el panel de operaciones se añadió un contador
GLOBAL de carritos activos, en un solo documento

  síntoma   en campaña, el contador se quedaba atrás y
            aparecían errores de contención
  causa     un documento admite del orden de una escritura
            por segundo sostenida
            y había hasta 340 modificaciones por segundo

corrección
  el contador se reparte en 100 documentos
  cada escritura elige uno al azar
  el panel suma los 100 al leer

  → y para el panel, un valor agregado cada 30 s desde el
    almacén analítico habría bastado y habría sido más
    barato                                    clase 236
  → se hizo así al final
```

Y los índices, recortados:

```
los documentos de sesión tenían 34 campos
indexados automáticamente                    34
consultados                                   4

tras excluir los 30 no consultados
  coste de escritura por documento        -71 %
  coste mensual de documentos      1.240 € → 380 €
```

Las reglas de acceso, probadas:

```
la aplicación móvil accede directamente a los documentos
de sesión
→ las reglas son el control de acceso, no hay servidor en
  medio
```

```
el simulador de reglas, con 12 casos
  ✓  usuario lee su propio documento
  ✓  usuario escribe su propio documento
  ✗  usuario lee el documento de OTRO usuario
     → la regla comparaba el identificador del documento,
       no el del usuario autenticado
     → cualquiera con un identificador podía leer
  ✓  usuario sin autenticar                    denegado
  ✗  usuario escribe un campo que no debería
     → la regla no validaba qué campos se modifican
  ✓  el resto                                  correcto

→ 2 de 12, y las dos exponían datos de otros usuarios
→ y sin el simulador no se habrían visto hasta que alguien
  las probara                                    ley 22
```

La telemetría en tabla ancha:

```
clave de fila   invertida(marca de tiempo) + sesión
  → el componente invertido reparte; sin él, todas las
    escrituras nuevas caerían juntas

grupos de columnas
  «evento» y «contexto» separados
  → el proceso en tiempo real lee solo «evento»

caducidad por celda                          30 días
  → y el histórico va al almacén analítico  clase 236

coste                                      610 €/mes
  frente a 3.900 € estimados con el bus de mensajes
                                             clase 224
```

El resultado:

```
coste mensual de datos                   6.200 €     3.320 €
  (según la propuesta inicial)
latencia p99 del inventario               890 ms       14 ms
caudal máximo del inventario            1.200/s      9.400/s
transacciones abortadas                    14 %        0,2 %
coste de documentos                      1.240 €       380 €
reglas de acceso con fallos                2/12         0/12
contadores con contención                     1           0
```

La lección que esta clase deja: la propuesta de poner todo en la base que escala habría costado casi el doble sin resolver ningún problema, porque el coste base de esas familias es alto y la mayoría de las cargas cabían en una relacional normal. Y el punto caliente apareció donde siempre: una clave autoincremental trasladada de la base relacional, que en un almacén repartido por rangos concentra el noventa y seis por ciento de las escrituras en un solo servidor.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-19-gcp-production-architecture/235-cloud-sql-spanner-firestore-y-bigtable/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 gcp-operational-data 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 gcp-operational-data 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
Latencia altísima y transacciones abortadas con la CPU baja	Clave monótona creciente: todas las escrituras nuevas caen en el mismo rango	Usa identificador aleatorio, invierte los bits o pon delante un componente que reparta; revisa las estadísticas de acceso por rango.
Las transacciones son lentas aunque toquen pocas filas	Cruzan servidores porque las tablas hijas no están entrelazadas con el padre	Entrelaza las hijas al crear la tabla; cambiarlo después exige recrear y migrar.
Se paga mucho por una carga pequeña	Se eligió una familia con coste base alto por si escala en el futuro	Compara el coste base antes de elegir; empieza en la relacional gestionada y registra qué señal justificaría migrar.
Un contador global da errores de contención	Un documento admite del orden de una escritura por segundo	Reparte el contador en varios documentos y suma al leer, o agrega el valor en el almacén analítico.
El coste de escritura de los documentos es desproporcionado	Se indexan automáticamente todos los campos	Excluye del índice los campos que no se consultan.
Un usuario puede leer los datos de otro	Las reglas de acceso comparan el identificador equivocado o no validan los campos modificados	Prueba las reglas con el simulador para todos los casos, incluidos los de acceso ajeno.

■ 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 cuatro familias?
2. ¿Por qué una clave autoincremental produce puntos calientes en un almacén repartido por rangos?
3. ¿Qué aporta entrelazar las tablas hijas y cuándo se decide?
4. ¿Qué límite tiene la escritura sobre un mismo documento y cómo se resuelve?
5. ¿Por qué no conviene empezar por la familia que más escala?

Referencias

- Google Cloud (2025). Spanner schema design and hotspot avoidance. <https://cloud.google.com/spanner/docs/schema-design>
- Google Cloud (2025). Spanner: interleaved tables and transactions. <https://cloud.google.com/spanner/docs/schema-and-data-model>
- Google Cloud (2025). Firestore: best practices and limits. <https://cloud.google.com/firestore/docs/best-practices>
- Google Cloud (2025). Bigtable schema design. <https://cloud.google.com/bigtable/docs/schema-design>
- Google Cloud (2025). Cloud SQL: read replicas and high availability. <https://cloud.google.com/sql/docs/mysql/high-availability>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 19 en PDF · Recorrido de Google Cloud en PDF · Manual integral

Anterior	Índice	Siguiente
← 234 · GKE Autopilot, Workload Identity y Config Sync	Parte 19 · Programa	236 · BigQuery, Dataflow, Dataproc y gobernanza de datos →

Evaluación — 235 Cloud SQL, Spanner, Firestore y Bigtable

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 gcp-operational-data 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.

236 — BigQuery, Dataflow, Dataproc y gobernanza de datos

Parte: 19 — Google Cloud: arquitectura de datos y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: data · Estado: EXECUTABLECORE

Propósito

Operar la plataforma analítica de Google Cloud, que es la pieza más madura de esta nube y la que más problemas produce de dos tipos concretos: coste, porque una consulta mal escrita puede costar cientos de euros en segundos, y gobierno, porque el acceso a datos personales se concede en la primera semana y nadie lo revisa después. La clase cubre el modelo de facturación, el diseño de tablas, el procesamiento por lotes y continuo, y el control de acceso a nivel de columna y de fila.

Resultados de aprendizaje

Al finalizar podrás:

1. Elegir el modelo de facturación adecuado y evitar las sorpresas.
2. Diseñar tablas particionadas y agrupadas para reducir lo que se lee.
3. Elegir entre procesamiento continuo, por lotes y gestionado.
4. Controlar el acceso a columnas y filas con datos sensibles.
5. Gobernar el linaje, la calidad y la retención de los datos.

Conceptos centrales

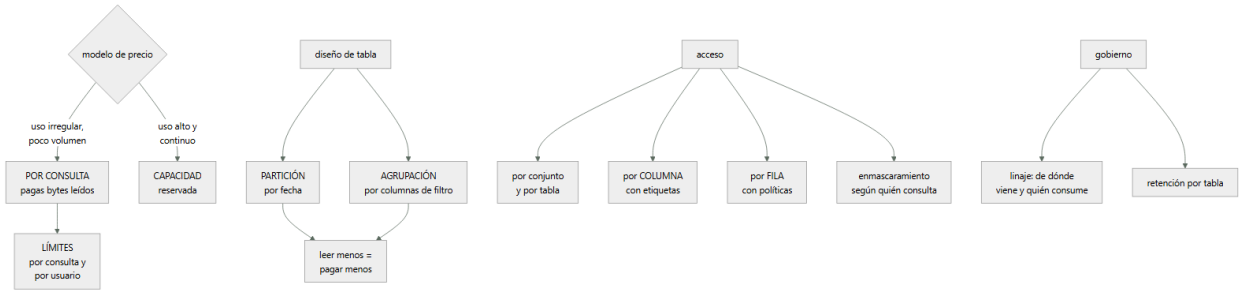
Concepto	Comprensión verificable
facturación por consulta	Se paga por bytes leídos. Una consulta sin filtro sobre una tabla grande cuesta mucho.
capacidad reservada	Modelo de precio por capacidad de proceso contratada, con coste previsible.
partición	División física de una tabla, normalmente por fecha. Permite leer solo lo necesario.
agrupación	Ordenación física por columnas dentro de la partición. Reduce lo leído en filtros por esas columnas.
enmascaramiento de columna	Transformación del valor según quién consulta, sin duplicar la tabla.
linaje	Registro de de dónde viene cada tabla y qué la consume. Necesario para retirar y para auditar.

Modelo mental

Google Cloud combina una jerarquía estricta de recursos con una red global y servicios data-first; proyecto, cuota e IAM forman una unidad operativa.

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 coste, que es lo primero

Aquí el coste no es un asunto de fin de mes: una consulta puede costar cientos de euros en segundos, y por eso se aborda antes que el diseño.

```
POR CONSULTA
  se paga por BYTES LEÍDOS, no por filas devueltas
  → «select * from tabla limit 10» sobre 40 TB lee 40 TB
  → el límite no reduce lo leído
  + sin coste cuando no se consulta
  – impredecible; una consulta mal escrita cuesta mucho

CAPACIDAD RESERVADA
  se paga por capacidad de proceso, con compromiso
  + coste previsible y consultas «gratis» dentro de la
    capacidad
  – hay que dimensionarla, y las consultas compiten entre sí

CRITERIO
  uso irregular o bajo   → por consulta, CON LÍMITES
  uso alto y continuo    → capacidad reservada
  y lo habitual: capacidad para lo productivo y por
  consulta para exploración, con límites
```

Y los límites que hay que poner el primer día:

```
LÍMITE DE BYTES POR CONSULTA
  «ninguna consulta puede leer más de X»
  → rechaza antes de ejecutar
  → y es lo único que impide el susto de una consulta

CUOTA DIARIA POR USUARIO Y POR PROYECTO

Y LA COSTUMBRE DE ESTIMAR ANTES
  la herramienta dice cuántos bytes leerá una consulta
  ANTES de ejecutarla
  → y esa estimación debe estar en la formación de
    cualquiera que consulte                    ley 26
```

Y las tres causas de coste que se repiten:

```
1  SELECCIONAR TODAS LAS COLUMNAS
   el almacenamiento es por columnas: leer 3 de 80 cuesta
   una fracción
   → «select *» es el error más caro y el más común

2  NO FILTRAR POR LA COLUMNA DE PARTICIÓN
   una consulta sin filtro de fecha lee la tabla entera
   → se puede EXIGIR el filtro en la definición de la tabla

3  CONSULTAS PROGRAMADAS QUE NADIE REVISA
   un panel que refresca cada 5 minutos sobre una tabla
   grande
   → y nadie mira su coste                     ley 15
```

2. Diseñar para leer menos

Todo el rendimiento y el coste dependen de cuántos bytes se leen. Hay dos mecanismos.

```
PARTICIÓN
  divide la tabla, normalmente por fecha
  → una consulta con filtro de fecha lee solo esas
    particiones

  y la opción que evita sustos
    exigir el filtro de partición en la tabla
    → una consulta sin él FALLA en vez de leer todo

AGRUPACIÓN
  ordena físicamente por hasta cuatro columnas dentro de
  la partición
  → filtrar por la primera columna de agrupación reduce
    mucho lo leído
  → el orden importa: se agrupa por lo que más se filtra

→ y las dos juntas son lo que convierte una tabla de 40 TB
  en consultas de 2 GB
```

Y el resto de decisiones de modelado:

```
DESNORMALIZAR
  las uniones cuestan; repetir datos suele salir más barato
  → aquí, al contrario que en una base transaccional

CAMPOS ANIDADOS Y REPETIDOS
  guardar las líneas de un pedido dentro del pedido
  → evita la unión y reduce lo leído

VISTAS MATERIALIZADAS
  agregados precalculados que se refrescan solos
  → las consultas los usan automáticamente
  → y reducen mucho el coste de los paneles

TABLAS EXTERNAS
  consultar ficheros del almacén sin cargarlos
  → cómodo, y más lento y más caro por consulta
  → para datos poco consultados
```

El almacenamiento, con su propia palanca:

```
las particiones que no se modifican pasan a un precio de
almacenamiento menor, solas
→ y la RETENCIÓN por partición retira lo antiguo
→ sin retención, la tabla crece indefinidamente   ley 25

y una decisión que hay que tomar
  ¿cuántos años de histórico hacen falta de verdad?
  → casi siempre menos de los que se guardan
  → y lo que haga falta por norma, en almacenamiento
    barato                                  clase 239
```

Y la carga de datos, con lo que decide el coste:

```
CARGA POR LOTES        gratuita o muy barata
INSERCIÓN CONTINUA     se paga por fila, y suma
ESCRITURA POR FLUJO    el mecanismo moderno, más barato

→ si el retraso aceptable es de minutos, la carga por lotes
  cada pocos minutos es mucho más barata que la inserción
  fila a fila
→ y esa es una decisión de arquitectura, no de
  implementación                              clase 242
```

3. Procesar: continuo, por lotes y gestionado

Hay tres formas de transformar datos y se eligen por lo que hay que hacer, no por preferencia.

```
CONSULTAS PROGRAMADAS Y VISTAS MATERIALIZADAS
  la transformación se escribe en SQL, dentro del almacén
  + lo más simple; sin infraestructura
  + y el linaje se registra solo
  – limitado a lo que SQL puede hacer
  → y es lo que resuelve la mayoría de los casos

PROCESAMIENTO UNIFICADO (Dataflow)
```

```
  el mismo código para lotes y para flujo continuo
  + gestiona ventanas, datos que llegan tarde y estado
  + escala solo
  − más que operar, y hay que entender el modelo
  → para transformaciones complejas o continuas de verdad

MOTOR DE CÓDIGO ABIERTO GESTIONADO (Dataproc)
  para trabajos existentes que ya están escritos
  + migración directa de lo que ya hay
  − hay que gestionar clústeres, aunque sean efímeros
  → y para trabajos nuevos, rara vez es la elección
```

Y el criterio:

```
¿se puede escribir en SQL y el retraso de minutos vale?
  → consulta programada o vista materializada
¿hace falta procesamiento continuo con ventanas y estado?
  → procesamiento unificado
¿hay trabajos que ya existen y funcionan?
  → motor gestionado, con fecha de revisión      ley 25
```

Y lo que hay que resolver en el procesamiento continuo, que es lo mismo de la clase 116:

```
DATOS QUE LLEGAN TARDE
  una ventana cerrada y un dato de hace 3 horas
  → decidir: descartar, reprocesar o corregir
  → y declararlo                              clase 242

EXACTAMENTE UNA VEZ
  se consigue con identificadores y deduplicación
  → no viene gratis                          clase 237

Y EL RETRASO
  vigilado, con alerta                         ley 13
  → «el flujo lleva N minutos de retraso» es la señal que
    dice que algo va mal antes de que falte un dato
```

Y la orquestación:

```
los trabajos dependen unos de otros
→ hace falta un orquestador que sepa el orden, reintente y
  avise                                      clase 243
→ y no un conjunto de trabajos programados por hora
  esperando que el anterior haya terminado
```

4. Gobierno: acceso, linaje y retención

El acceso a datos sensibles es donde este tipo de plataformas produce los hallazgos más incómodos.

```
LOS NIVELES DE CONTROL
  por conjunto de datos y por tabla
    → el habitual, y el más grueso
  POR COLUMNA
    las columnas se etiquetan; el acceso se concede por
    etiqueta
    → «quien no tenga la etiqueta de datos personales no ve
      esas columnas»
  POR FILA
    políticas que filtran según quién consulta
    → «cada país ve sus propias filas»
  ENMASCARAMIENTO
    la columna se transforma según quién consulta
    → un analista ve el correo con formato pero sin valor
    → sin duplicar la tabla
```

Y el problema que resuelven, que es el de la clase 230:

```
sin control por columna, dar acceso a una tabla es dar
acceso a CUANTO contiene
→ y por eso el acceso se concede al conjunto entero
→ y por eso los equipos de análisis acaban viendo datos
  personales completos que no necesitan

con etiquetas de columna
  el acceso a la tabla no incluye las columnas etiquetadas
  → y la mayoría de los análisis funcionan igual
```

Y la disciplina:

```
1  CLASIFICAR las columnas: la plataforma puede detectar
   datos personales automáticamente
2  ETIQUETAR las sensibles
3  QUITAR el acceso amplio y conceder por etiqueta
4  y REVISAR trimestralmente quién la tiene   clase 230
```

El linaje, que hace posible retirar y auditar:

```
qué tablas alimentan a cuáles
qué consultas y paneles consumen cada tabla
y de dónde vino cada dato

→ sin linaje, retirar una tabla es imposible: nadie sabe
  quién la usa                                    ley 20
→ y ante una petición de borrado de datos personales,
  tampoco se sabe dónde se copiaron   clases 139, 251
```

Y lo que hay que vigilar en esta plataforma:

```
coste por consulta, por usuario y por panel
consultas que leen más de X bytes
tablas sin partición ni agrupación
tablas sin retención
tablas sin consumidores en 90 días        → candidatas a
                                            retirar
acceso a columnas sensibles, por identidad
retraso de los flujos continuos
y trabajos que fallan sin que nadie lo mire      ley 13
```

Y la lista de comprobación de la clase:

- hay límite de bytes por consulta
- hay cuota diaria por usuario y por proyecto
- las tablas grandes están particionadas
- se exige el filtro de partición donde procede
- están agrupadas por las columnas de filtro habituales
- los paneles usan vistas materializadas
- hay retención por tabla
- la ingesta usa lotes o escritura por flujo, no inserción fila a fila
- las columnas sensibles están clasificadas y etiquetadas
- el acceso se concede por etiqueta, no al conjunto entero
- hay políticas de fila donde aplica
- el linaje está activado y se consulta
- hay alerta de retraso de los flujos y de trabajos fallidos
- se revisan las tablas sin consumidores

Y el cierre que enlaza con la clase siguiente: con datos operativos y analíticos resueltos, queda la mensajería que los conecta, que en esta nube tiene una propiedad que se anuncia mucho y conviene entender bien. Es la materia de la clase 237.

Ejemplo trabajado

CloudShop monta su plataforma analítica en Google Cloud. Lo que sigue son las tres consultas que costaban 4.100 € al mes, el acceso a datos personales que tenían 41 personas, y el rediseño de tablas que redujo el coste un 88 %.

La factura, al revisar:

```
coste mensual del almacén analítico          7.900 €
  consultas                                  6.200 €
  almacenamiento                             1.400 €
  inserción continua                           300 €

las consultas, por origen
  paneles programados                        4.100 €  ←
  exploración de analistas                   1.400 €
  transformaciones programadas                 700 €
```

Y el desglose de los paneles:

```
panel                    refresco   bytes/ejec.   €/mes
ventas en tiempo real      5 min       2,1 TB    2.900
```

```
ocupación por almacén      15 min       0,9 TB      810
embudo de conversión        1 h         3,4 TB      390

→ tres consultas, 4.100 €/mes
→ y los tres paneles los abría alguien 2 o 3 veces al día
```

Y el diagnóstico de la primera:

```
la consulta del panel de ventas
  select * from pedidos p join lineas l ... where
  fecha >= current_date() - 30

  la tabla pedidos                          14 TB
  la tabla lineas                           26 TB
  ninguna estaba particionada
  → el filtro de fecha no reducía nada: se leía todo
  → y «select *» traía 80 columnas de las que se usaban 6

  bytes leídos por ejecución                 2,1 TB
  ejecuciones al mes                         8.640
```

El rediseño de tablas:

```
pedidos
  partición    por fecha de pedido, diaria
  agrupación   por país, canal, estado
  filtro de partición    EXIGIDO

lineas
  partición    por fecha del pedido (heredada)
  agrupación   por producto
  o mejor: las líneas ANIDADAS dentro de pedidos
  → se eligió anidarlas: elimina la unión

y las consultas reescritas
  columnas explícitas, no «select *»
  filtro de fecha, obligatorio

bytes por ejecución del panel de ventas   2,1 TB → 1,4 GB
```

Y las vistas materializadas:

```
los tres paneles consultaban agregados diarios
→ vista materializada con el agregado, refrescada sola
→ los paneles la usan automáticamente

bytes por ejecución                      1,4 GB → 12 MB

coste de los tres paneles          4.100 € → 41 €/mes
```

Los límites, puestos después del susto:

```
un analista ejecutó una consulta exploratoria sin filtro
sobre la tabla de eventos de 61 TB
  bytes leídos                              61 TB
  coste de esa consulta                     310 €
  duración                                   94 s

→ y no había nada que lo impidiera

límites puestos
  bytes máximos por consulta                 200 GB
    → salvo excepción con aprobación
  cuota diaria por usuario                   2 TB
  cuota diaria por proyecto de exploración   20 TB

y formación
  la estimación previa de bytes, en la guía de
  incorporación
  → y un aviso en la herramienta cuando la estimación
    supera 50 GB

consultas rechazadas por límite, primer mes          61
  de ellas, errores reales del analista              57
  con excepción aprobada                              4
```

El acceso a datos personales.

```
al inventariar
  personas con acceso al conjunto de datos de clientes  41
  de ellas, que necesitaban ver datos personales
  completos                                               3

  → los otros 38 hacían análisis agregados y no
    necesitaban el correo, el teléfono ni la dirección
  → pero el acceso se concede al conjunto, y el conjunto
    los incluye                                clase 230

y la clasificación automática encontró
  columnas con datos personales                          61
    en tablas donde nadie sabía que los había             9
      · una tabla de análisis de campañas con correos
      · dos tablas temporales de una migración de 2023
        que nunca se borraron                    ley 25
```

Y la corrección:

```
clasificación y etiquetado de las 61 columnas
etiquetas jerárquicas
  personal-directo    (nombre, correo, teléfono,
                      dirección)
  personal-indirecto  (identificador, dispositivo)
  financiero

acceso concedido por etiqueta, no por conjunto
  personal-directo                    3 personas
  personal-indirecto                 12 personas
  financiero                          6 personas
  el resto del conjunto              41 personas

y enmascaramiento para los casos intermedios
  el correo se ve como «a***@dominio.com»
  → suficiente para verificar y sin exponer el valor

y las políticas de fila
  el equipo de cada país ve solo las filas de su país

y las 3 tablas con datos que nadie sabía
  2 borradas (temporales de la migración)
  1 seudonimizada
```

Y la comprobación:

```
prueba negativa
  un analista sin la etiqueta consulta la columna de
  correo
  → la consulta falla, indicando qué etiqueta falta
  un analista de España consulta filas de Portugal
  → devuelve 0 filas
  → y ambas se ejecutan en cada despliegue del gobierno
                                                ley 22
```

El procesamiento, ajustado:

```
antes
  inserción continua fila a fila desde la aplicación
  41.000 eventos/s → 300 €/mes de inserción
  y transformaciones en un motor gestionado con un clúster
  siempre encendido                          410 €/mes

después
  los eventos van al servicio de mensajería y de ahí en
  lotes cada 60 s                          clase 237
    coste de ingesta                    300 € → 18 €
  las transformaciones que caben en SQL, como consultas
    programadas                                    9
  las que no, en procesamiento unificado           2
    clúster gestionado retirado          410 € → 140 €

y el retraso declarado
  eventos disponibles para consulta      ≤ 90 s
```

```
  con alerta si supera 5 min                    ley 13
```

La retención y el linaje:

```
retención por tabla
  eventos brutos                        90 días
  agregados diarios                      5 años
  tablas de trabajo                      7 días

  almacenamiento                 1.400 € → 310 €

linaje activado
  tablas sin consumidores en 90 días                34
    → 28 borradas tras avisar
    →  6 resultaron alimentar informes trimestrales
  y ante la primera petición de borrado de datos de un
  cliente
    tablas donde aparecía, según el linaje             7
    tablas que el equipo creía                         2
                                        clases 139, 251
```

El resultado:

```
coste mensual del almacén                7.900 €       610 €
  paneles                                4.100 €        41 €
  exploración                            1.400 €       120 €
  almacenamiento                         1.400 €       310 €
  ingesta                                  300 €        18 €
bytes por ejecución del panel de ventas    2,1 TB      12 MB
personas con acceso a datos personales        41          3
columnas sensibles sin clasificar             61          0
tablas con datos personales desconocidos       3          0
tablas sin consumidores                       34          6
consultas que pueden leer la tabla entera  cualquiera      0
```

La lección que esta clase deja: tres paneles que alguien abría dos veces al día costaban cuatro mil cien euros al mes, y el problema no era la plataforma sino tablas sin particionar consultadas con «select ». Y cuarenta y una personas tenían acceso a datos personales completos porque el acceso se concede al conjunto y el conjunto los incluye; solo tres los necesitaban, y el resto siguió trabajando igual con las columnas enmascaradas.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-19-gcp-production-architecture/236-bigquery-dataflow-dataproc-y-gobernanza-de-datos/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 gcp-analytics-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 gcp-analytics-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
Una consulta cuesta cientos de euros en segundos	Se paga por bytes leídos y no hay límite ni filtro de partición	Pon límite de bytes por consulta y cuota por usuario, particiona las tablas y exige el filtro de partición.
Los paneles dominan la factura	Refrescan consultas pesadas sobre tablas grandes muchas veces al día	Usa vistas materializadas con los agregados y revisa el coste por panel.
El filtro de fecha no reduce el coste	La tabla no está particionada por esa columna	Particiona por la columna de filtro habitual y agrupa por las siguientes más usadas.
Todo el equipo de análisis ve datos personales completos	El acceso se concede al conjunto de datos, que los incluye	Clasifica y etiqueta las columnas sensibles, concede por etiqueta y usa enmascaramiento para los casos intermedios.
La ingesta continua cuesta más que el proceso	Se inserta fila a fila cuando el retraso aceptable es de minutos	Agrupa en lotes o usa la escritura por flujo; decide el retraso aceptable como decisión de arquitectura.
No se puede retirar una tabla porque nadie sabe quién la usa	No hay linaje registrado	Activa el linaje y revisa periódicamente las tablas sin consumidores.

■ 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 un límite de filas no reduce el coste de una consulta?
2. ¿Qué diferencia hay entre partición y agrupación, y para qué sirve cada una?
3. ¿Qué problema resuelve el control de acceso por columna?
4. ¿Cuándo conviene procesamiento unificado frente a consultas programadas?
5. ¿Para qué hace falta el linaje además de para auditar?

Referencias

- Google Cloud (2025). BigQuery: controlling costs. <https://cloud.google.com/bigquery/docs/best-practices-costs>
- Google Cloud (2025). Partitioned and clustered tables. <https://cloud.google.com/bigquery/docs/partitioned-tables>
- Google Cloud (2025). Column-level and row-level security. <https://cloud.google.com/bigquery/docs/column-level-security-intro>
- Google Cloud (2025). Dataflow: unified batch and streaming. <https://cloud.google.com/dataflow/docs/concepts>

- Google Cloud (2025). Dataplex and data lineage. <https://cloud.google.com/dataplex/docs/about-data-lineage>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 19 en PDF · Recorrido de Google Cloud en PDF · Manual integral

Anterior	Índice	Siguiente
← 235 · Cloud SQL, Spanner, Firestore y Bigtable	Parte 19 · Programa	237 · Pub/Sub, Eventarc y entrega exactamente-una-vez →

Evaluación — 236 BigQuery, Dataflow, Dataproc y gobernanza de datos

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 gcp-analytics-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.

237 — Pub/Sub, Eventarc y entrega exactamente-una-vez

Parte: 19 — Google Cloud: arquitectura de datos y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: messaging · Estado: EXECUTABLECORE

Propósito

Montar la mensajería de Google Cloud, cuyo servicio principal anuncia entrega exactamente una vez, y conviene entender con precisión qué significa y qué sigue siendo responsabilidad del código. La clase separa el servicio de mensajería del enrutador de eventos, explica los mecanismos de entrega y sus parámetros, y sostiene lo que este programa lleva demostrando: la garantía del transporte no elimina la necesidad de idempotencia en el efecto.

Resultados de aprendizaje

Al finalizar podrás:

1. Elegir entre suscripción de entrega y de extracción, y el enrutador de eventos.
2. Configurar plazos, reintentos y destino de fallidos coherentemente.
3. Entender qué garantiza la entrega exactamente una vez y qué no.
4. Ordenar mensajes por clave cuando el orden importa.
5. Operar los fallidos con alerta, diagnóstico y reproceso probado.

Conceptos centrales

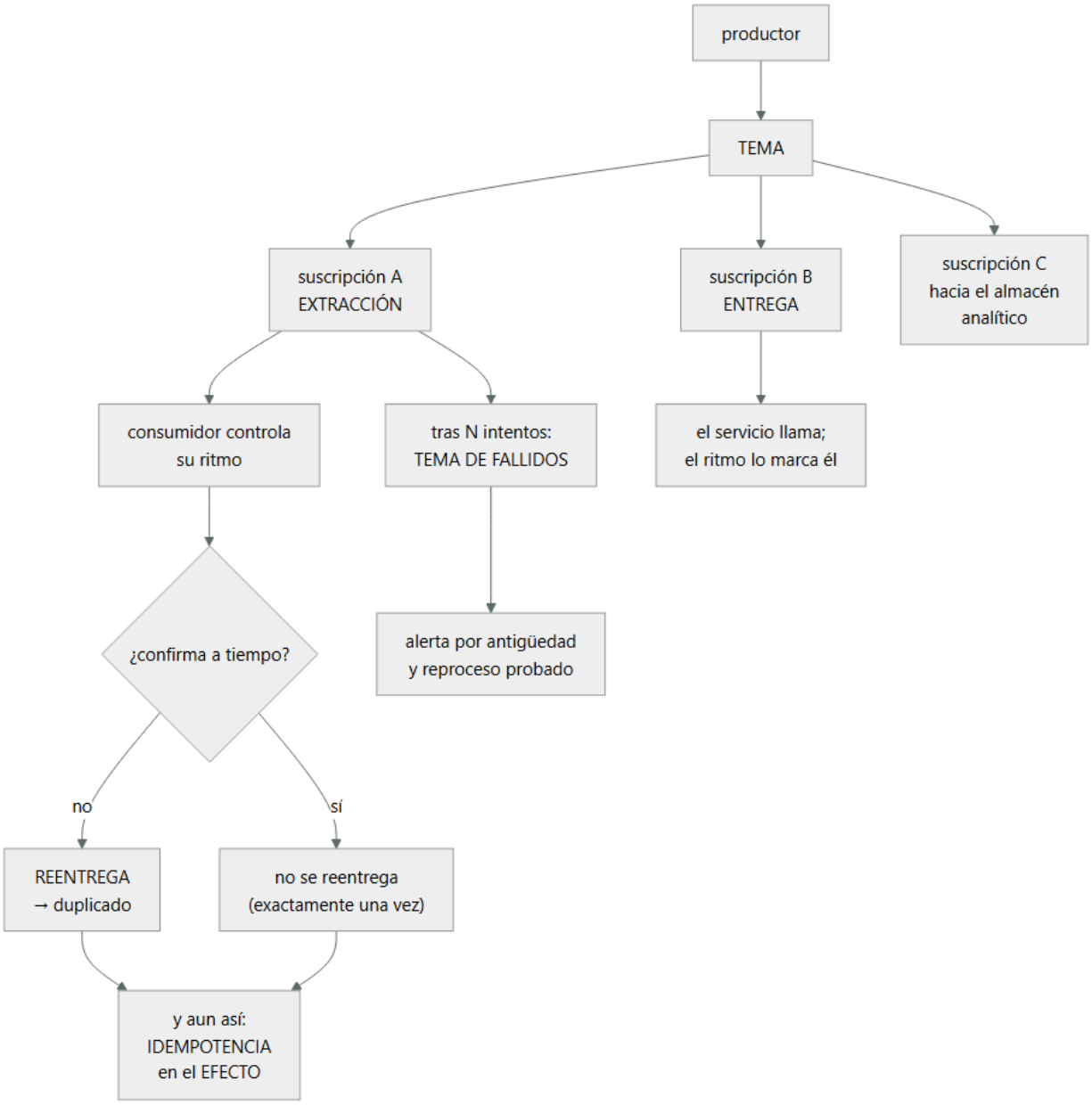
Concepto	Comprensión verificable
tema	Punto de publicación. Los consumidores se conectan mediante suscripciones independientes.
suscripción de extracción	El consumidor pide mensajes. Controla su propio ritmo.
suscripción de entrega	El servicio llama a un punto del consumidor. Más simple y con menos control de ritmo.
plazo de confirmación	Tiempo que el mensaje queda retenido tras entregarse. Si vence, se reentrega.
entrega exactamente una vez	Garantía de que un mensaje confirmado no se reentrega dentro de la ventana de confirmación.
clave de ordenación	Campo que agrupa mensajes para entregarlos en orden. Limita el caudal por clave.

Modelo mental

Google Cloud combina una jerarquía estricta de recursos con una red global y servicios data-first; proyecto, cuota e IAM forman una unidad operativa.

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 tema, muchas suscripciones

El modelo aquí es más simple que el de las otras nubes: hay un servicio principal y un enrutador.

```
SERVICIO DE MENSAJERÍA
  un TEMA donde se publica
  N SUSCRIPCIONES independientes, cada una con su propio
  ritmo, sus reintentos y su destino de fallidos
  → uno a muchos, sin configurar nada más
  → y añadir un consumidor es crear una suscripción: el
    productor no se entera                    clase 148

ENRUTADOR DE EVENTOS
  entrega eventos de la propia plataforma y de fuentes
  propias a destinos, con filtros
  → «cuando se suba un fichero a este almacén, llama a este
    servicio»
  → y por debajo usa el servicio de mensajería
```

Y los dos tipos de suscripción, que se eligen mal a menudo:

```
EXTRACCIÓN
```

```
  el consumidor pide mensajes cuando puede
  + controla su ritmo: no le llega más de lo que aguanta
  + amortigua picos de forma natural
  − hay que ejecutar un proceso que extraiga

ENTREGA (a un punto HTTP)
  el servicio llama al consumidor
  + más simple: no hay proceso extractor
  + encaja con servicios que escalan por peticiones
                                                clase 233
  − el ritmo lo marca el servicio, no el consumidor
  − y por tanto puede escalar el consumidor sin control

Y LA DECISIÓN
  si el consumidor tiene un recurso limitado detrás (una
  base), la extracción da control
  si escala bien y es idempotente, la entrega es más simple
  → y en cualquier caso hay que poner un límite de
    concurrencia                          clase 233
```

Y los otros destinos, que ahorran código:

```
SUSCRIPCIÓN HACIA EL ALMACÉN ANALÍTICO
  los mensajes se escriben directamente en una tabla
  → sin consumidor que mantener
  → y con esquema validado                  clase 236

SUSCRIPCIÓN HACIA ALMACENAMIENTO
  vuelca por lotes a ficheros
  → el histórico, sin escribir nada         clase 224
```

2. Exactamente una vez: qué significa

Esta es la propiedad que más se malinterpreta, y conviene ser preciso.

```
LO QUE GARANTIZA
  un mensaje CONFIRMADO no se vuelve a entregar dentro de
  la ventana de confirmación
  → elimina los duplicados que produce el plazo vencido
    cuando el consumidor sí había terminado

LO QUE NO GARANTIZA
  que el PRODUCTOR no publique el mismo hecho dos veces
    → un reintento del productor crea dos mensajes
      distintos
  que el efecto no se produzca dos veces si el consumidor
    falla DESPUÉS de actuar y ANTES de confirmar
  que el reproceso desde el tema de fallidos no repita
  ni que dos suscripciones no procesen lo mismo

→ es decir: elimina una de las cinco fuentes de duplicado
  de la clase 210, no las cinco
```

Y la conclusión, que es la de siempre:

```
LA IDEMPOTENCIA EN EL EFECTO SIGUE SIENDO OBLIGATORIA
  clave del suceso, generada por el productor
  registro de claves procesadas, en dos estados
  y escritura condicionada antes del efecto  clase 210

→ lo que la garantía da es que se necesitan MENOS
  reintentos y menos reproceso, no que se pueda prescindir
  de la comprobación
```

Y el coste, que conviene conocer:

```
la entrega exactamente una vez tiene un coste
  el caudal por suscripción es menor
  la latencia es algo mayor
  y hay restricciones de configuración

→ activarla donde el duplicado cuesta dinero
→ y no en la telemetría, donde no aporta y limita el caudal
```

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

```
PLAZO DE CONFIRMACIÓN
  tiempo que el mensaje queda retenido tras entregarse
  si vence antes de confirmar, se reentrega

  y la extensión automática
    las bibliotecas extienden el plazo mientras se procesa
    → activarla es lo que evita el problema
    → y hay un máximo total: procesos muy largos no encajan

CONCURRENCIA DEL CONSUMIDOR
  cuántos mensajes procesa a la vez
  → si es alta y el proceso es lento, se acumulan mensajes
    retenidos y algunos vencen
  → es la misma relación de prebúsqueda de la clase 224

REINTENTOS Y RETROCESO
  con retroceso exponencial declarado
  → sin retroceso, un consumidor caído recibe una avalancha
    en cuanto vuelve
```

3. Orden, filtros y fallidos

El orden se consigue con clave, y tiene su coste.

```
CLAVE DE ORDENACIÓN
  los mensajes con la misma clave se entregan en orden
  → y en la misma suscripción, al mismo consumidor

  coste
    el caudal por clave lo limita un solo consumidor
    y si un mensaje falla, los siguientes de esa clave
    ESPERAN
    → un mensaje venenoso bloquea su clave entera

  y hay que decidir
    ¿qué se hace con el veneno? ¿se aparta y se sigue, o se
    para?
    → misma decisión que en la clase 224
```

Y la pregunta previa, que evita el problema:

```
¿de verdad hace falta orden, o basta con que las
operaciones sean conmutativas?
  «−1 unidad» no necesita orden
  «estado = ENVIADO» tampoco, si lleva versión
  «stock = 4» sí
→ y modelar para no necesitar orden es más barato que
  garantizarlo                          clases 203, 224
```

Los filtros, que reducen trabajo y coste:

```
una suscripción puede filtrar por atributos del mensaje
  → el consumidor solo recibe lo suyo
  → y no se paga por entregar lo que se iba a descartar

y por eso conviene que los atributos lleven lo necesario
para filtrar
  tipo de suceso, entidad, entorno, versión de esquema
```

El tema de fallidos, con la disciplina de siempre:

```
tras N intentos, el mensaje va a otro tema
  → y hay que crear una suscripción sobre él, o los
    mensajes caducan sin que nadie los vea         ley 13

LAS ALERTAS
  «hay mensajes sin confirmar con más de 1 hora de
   antigüedad»                              ← la que sirve
  «el número de mensajes sin confirmar crece de forma
   sostenida»
  y en el tema de fallidos, las dos                ley 15

EL DIAGNÓSTICO
  guardar el motivo del fallo con el mensaje
  → veneno, dependencia o error de código   clase 210
```

```
EL REPROCESO
  procedimiento escrito, con límite de ritmo y por lotes
  y probado por alguien que no lo escribió          ley 22
```

Y una capacidad que evita reprocesos:

```
RETROCESO EN EL TIEMPO
  una suscripción puede volver a un momento anterior o a
  una instantánea
  → y reprocesar desde ahí sin mover mensajes
  → útil cuando un consumidor procesó mal durante un rato
  → y exige que el consumidor sea idempotente, otra vez
```

4. Operar y medir

Lo que hay que vigilar, con las señales que dicen algo:

```
MENSAJES SIN CONFIRMAR
  número y ANTIGÜEDAD del más viejo
  → la antigüedad es la que detecta el consumidor parado

PLAZOS VENCIDOS
  señal directa de plazo mal puesto o proceso lento

RETRASO DE PUBLICACIÓN A CONSUMO
  el tiempo entre que ocurre el hecho y se procesa
  → es la medida que le importa al negocio    clase 238

MENSAJES EN EL TEMA DE FALLIDOS, y su antigüedad

ENTREGAS REPETIDAS del mismo mensaje
  → si la entrega exactamente una vez está activa y aun así
    hay repeticiones, algo no está confirmando

Y EL COSTE
  se paga por volumen publicado, entregado y almacenado
  → una suscripción que nadie consume acumula y factura
  → suscripciones sin consumidor: inventario periódico
                                                    ley 25
```

Y una advertencia de coste que se repite:

```
los mensajes se retienen hasta que se confirman o caducan
  → una suscripción abandonada retiene todo hasta la
    caducidad
  → y factura almacenamiento
→ y por eso las suscripciones sin consumidor son un
  hallazgo típico del primer inventario
```

Las comprobaciones de esta clase:

- enviar el mismo mensaje 50 veces y comprobar un efecto
- parar el consumidor y esperar la alerta de antigüedad
- dejar un mensaje en el tema de fallidos y esperar alerta
- reprocesar 10.000 mensajes con límite de ritmo
- publicar un mensaje con esquema inválido
- retroceder una suscripción y comprobar el reproceso
- y comprobar que un mensaje venenoso con clave de ordenación no bloquea indefinidamente

Y los errores de traslado, que esta parte vigila:

```
✗ «como garantiza exactamente una vez, no hace falta
   idempotencia»
  → cubre una de las cinco fuentes de duplicado

✗ «la suscripción de entrega es como una cola con
   consumidor»
  → el ritmo lo marca el servicio; hay que limitar la
    concurrencia del consumidor

✗ «un tema por consumidor»
  → un tema, N suscripciones; crear un tema por consumidor
    obliga al productor a saber quién escucha  clase 148
```

Y la lista de comprobación de la clase:

- hay un tema y una suscripción por consumidor
- el tipo de suscripción corresponde al consumidor
- la concurrencia del consumidor está limitada
- el plazo de confirmación supera el proceso, o se extiende
- los reintentos tienen retroceso declarado
- la entrega exactamente una vez está donde el duplicado cuesta
- toda operación con efecto es idempotente igualmente
- el orden por clave se usa solo donde hace falta
- está decidido qué hacer con un veneno en una clave ordenada
- los filtros reducen lo que llega a cada consumidor
- hay tema de fallidos CON suscripción
- hay alerta por antigüedad, no solo por número
- el motivo del fallo se guarda con el mensaje
- el reproceso está escrito y probado
- no hay suscripciones sin consumidor

Y el cierre que enlaza con la clase siguiente: con datos, cómputo y mensajería en pie, falta saber qué ocurre cuando algo va mal. La observabilidad de esta nube, con el estándar abierto, es la materia de la clase 238.

Ejemplo trabajado

CloudShop monta la mensajería de su plataforma en Google Cloud. Lo que sigue es el malentendido sobre la entrega exactamente una vez, la suscripción abandonada que acumuló 41 millones de mensajes, y la clave de ordenación que bloqueó un cliente durante 6 horas.

El malentendido, semana 2.

```
el equipo activó la entrega exactamente una vez en las 6
suscripciones
y retiró la comprobación de idempotencia de los
consumidores
motivo   «ya no hace falta: el servicio lo garantiza»

lo que pasó en la primera campaña
  correos de confirmación duplicados              214
  descuentos de stock duplicados                   88

el diagnóstico
  la garantía cubre: un mensaje CONFIRMADO no se reentrega
  no cubría lo que estaba pasando

  caso 1 (correos)
    el consumidor enviaba el correo y fallaba al confirmar
    por una desconexión
    → el mensaje se reentregaba, correctamente
    → y el correo se enviaba otra vez
    → la garantía funcionó: el problema es que el EFECTO ya
      se había producido

  caso 2 (stock)
    el productor reintentaba al no recibir confirmación de
    publicación
    → publicaba DOS mensajes distintos, con contenido igual
    → dos mensajes distintos, entregados una vez cada uno
    → la garantía no aplica: son mensajes distintos

corrección
  idempotencia restaurada, con clave del suceso generada
  por el PRODUCTOR y estable entre reintentos  clase 210
  → y la entrega exactamente una vez se dejó activa donde
    el duplicado cuesta, y se desactivó en telemetría,
    donde limitaba el caudal

duplicados tras la corrección                      0
```

Y la frase que el equipo escribió en el registro de decisión:

```
«la garantía del transporte reduce los duplicados de
 transporte; el efecto lo protege el consumidor»
→ y es la misma conclusión de las clases 117, 210 y 224,
  con otro producto                              ley 18
```

La suscripción abandonada.

```
al revisar el coste, mes 4
  coste de mensajería                        1.840 €/mes
  esperado                                     ~200 €

desglose
  publicación                                    140 €
  entrega                                        180 €
  ALMACENAMIENTO DE MENSAJES SIN CONFIRMAR     1.520 €  ←

causa
  una suscripción creada para una prueba en el mes 1
  su consumidor se apagó al terminar la prueba
  la suscripción siguió recibiendo CUANTO se publicaba
  mensajes retenidos                          41 millones
  antigüedad del más antiguo                      94 días
  → y la retención estaba en 7 días… en OTRA suscripción

  la abandonada tenía retención de 31 días y los mensajes
  se acumulaban hasta caducar y volver a acumularse

corrección
  suscripción borrada
  inventario de suscripciones sin actividad de consumo
    → 4 más encontradas
  alerta: «suscripción sin consumo en 24 h»
  y caducidad automática de suscripciones inactivas 30 días
                                                     ley 25

coste                            1.840 € → 190 €/mes
```

La clave de ordenación que bloqueó un cliente.

```
el flujo de integración con socios usaba clave de
ordenación = identificador de socio
motivo   los eventos de un socio deben procesarse en orden

qué pasó
  un socio envió un evento con un campo con un carácter que
  el consumidor no manejaba
  → el consumidor fallaba al procesarlo
  → reintentaba con retroceso
  → y todos los eventos siguientes de ESE socio esperaban

  duración                                        6 h 10
  eventos acumulados de ese socio                  4.100
  otros socios                               sin afectación

cómo se detectó
  el socio llamó
  → la alerta de antigüedad existía pero con umbral de
    24 h                                          ley 15

correcciones
  1  umbral de antigüedad a 30 minutos
  2  decisión escrita sobre el veneno en clave ordenada
     tras 5 intentos, el mensaje se aparta al tema de
     fallidos y la clave AVANZA
     → se pierde el orden estricto para esa clave, y se
       registra como incidencia de datos
     → la alternativa (parar) era peor para el negocio
  3  validación de esquema en la publicación
     → el mensaje con el carácter problemático se habría
       rechazado al publicar                     clase 188
```

La configuración final:

```
consumidor          tipo         plazo   concurr.  orden
notificaciones      entrega       60 s     50       no
inventario          extracción   600 s     20       no
facturación         extracción   600 s     10       no
integración socios  extracción   300 s      5       SÍ
analítica           hacia el almacén analítico, directa
```

```
histórico           hacia almacenamiento, por lotes

entrega exactamente una vez
  activa en inventario, facturación e integración
  inactiva en notificaciones y telemetría

filtros
  cada suscripción filtra por tipo de suceso
  → la de inventario recibía 41.000 mensajes/día y ahora
    2.100
  → y no se paga por los 38.900 que descartaba

temas de fallidos
  uno por suscripción, CON suscripción propia
  con el motivo guardado como atributo
```

Y las dos suscripciones directas, que ahorraron trabajo:

```
ANALÍTICA
  antes   un consumidor que leía y escribía en el almacén
          analítico: 340 líneas y un servicio que mantener
  después suscripción directa hacia la tabla, con esquema
          validado                              clase 236
  → servicio retirado                             ley 23

HISTÓRICO
  antes   un trabajo que exportaba cada noche
  después suscripción hacia almacenamiento, por lotes de
          5 minutos
  → trabajo retirado
```

Las comprobaciones, ejecutadas:

```
✓  mismo mensaje 50 veces                    1 efecto
✓  parar el consumidor → alerta de antigüedad    3 min
✓  mensaje en tema de fallidos → alerta           41 s
✓  reprocesar 10.000 con límite de ritmo       2 min
✗  publicar un mensaje con esquema inválido
   → se aceptó: la validación de esquema no estaba
     activada en 2 de los 6 temas
   → activada
✓  retroceder una suscripción y reprocesar
✗  veneno con clave de ordenación
   → bloqueó indefinidamente en la primera prueba
   → corregido con la decisión de apartar y avanzar
```

El resultado:

```
coste de mensajería                     1.840 €       190 €
duplicados por campaña                     302           0
suscripciones sin consumidor                 5           0
mensajes retenidos                       41 M          <10 k
bloqueo por veneno en clave ordenada      6 h 10      6 min
tiempo de detección de consumidor parado    24 h       3 min
servicios de integración retirados             —           2
mensajes entregados y descartados       38.900/día        0
```

La lección que esta clase deja: la garantía de entrega exactamente una vez es real y cubre una de las cinco fuentes de duplicado; quitar la idempotencia porque existe produjo trescientos dos duplicados en una campaña. Y el mayor coste del capítulo no fue de proceso ni de entrega: fue mil quinientos veinte euros al mes de almacenamiento de una suscripción que nadie consumía desde hacía tres meses.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-19-gcp-production-architecture/237-pub-sub-eventarc-y-entrega-exactamente-una-vez/
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 gcp-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 gcp-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
Aparecen duplicados pese a tener entrega exactamente una vez	La garantía cubre la reentrega tras confirmar, no el reintento del productor ni el fallo tras producir el efecto	Mantén la idempotencia en el consumidor con clave del suceso generada por el productor y estable entre reintentos.
La factura de mensajería es diez veces lo previsto	Una suscripción sin consumidor retiene todo lo publicado y paga almacenamiento	Inventaría suscripciones sin consumo, alerta cuando una lleve 24 h sin consumir y caduca las inactivas.
Los eventos de un cliente se detienen durante horas	Un mensaje venenoso bloquea su clave de ordenación	Decide y escribe qué ocurre con el veneno: apartarlo y avanzar, o parar; valida el esquema al publicar.
El consumidor recibe más carga de la que aguanta	Suscripción de entrega, donde el ritmo lo marca el servicio	Limita la concurrencia del consumidor o usa suscripción de extracción si hay un recurso limitado detrás.
Vencen los plazos y se reentregan mensajes que sí se estaban procesando	El plazo de confirmación es menor que el proceso y no se extiende	Activa la extensión automática del plazo y reduce la concurrencia; vigila los plazos vencidos.
Los mensajes fallidos caducan sin que nadie los vea	El tema de fallidos no tiene suscripción ni alerta	Crea suscripción sobre él, alerta por antigüedad y guarda el motivo del fallo como atributo.

■ 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 exactamente la entrega exactamente una vez y qué no?
2. ¿Cuándo conviene suscripción de extracción y cuándo de entrega?
3. ¿Qué coste tiene usar clave de ordenación y qué hay que decidir?
4. ¿Por qué una suscripción sin consumidor genera coste?
5. ¿Qué dos suscripciones directas evitan escribir un consumidor?

Referencias

- Google Cloud (2025). Pub/Sub: exactly-once delivery. <https://cloud.google.com/pubsub/docs/exactly-once-delivery>
- Google Cloud (2025). Pub/Sub subscription types and delivery. <https://cloud.google.com/pubsub/docs/subscriber>
- Google Cloud (2025). Message ordering. <https://cloud.google.com/pubsub/docs/ordering>
- Google Cloud (2025). Dead-letter topics and retry policies. <https://cloud.google.com/pubsub/docs/handling-failures>
- Google Cloud (2025). Eventarc. <https://cloud.google.com/eventarc/docs/overview>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 19 en PDF · Recorrido de Google Cloud en PDF · Manual integral

Anterior	Índice	Siguiente
← 236 · BigQuery, Dataflow, Dataproc y gobernanza de datos	Parte 19 · Programa	238 · Cloud Operations, Trace y OpenTelemetry →

Evaluación — 237 Pub/Sub, Eventarc y entrega exactamente-una-vez

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 gcp-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.

238 — Cloud Operations, Trace y OpenTelemetry

Parte: 19 — Google Cloud: arquitectura de datos y operación en producción Nivel: avanzado · Horas estimadas: 4
Laboratorio: observability · Estado: EXECUTABLECORE

Propósito

Montar la observabilidad de Google Cloud, donde los registros de auditoría son la pieza distintiva y a la vez la que más coste y más sorpresas produce: el registro de acceso a datos está desactivado por defecto, y activarlo sin acotar dispara la factura. La clase cubre el enrutado y los filtros de exclusión, la instrumentación estándar, los objetivos de nivel de servicio como recurso declarado, y las alertas que este programa exige siempre.

Resultados de aprendizaje

Al finalizar podrás:

1. Enrutar registros con filtros de inclusión y exclusión, y controlar su coste.
2. Decidir qué registros de auditoría activar y sobre qué servicios.
3. Instrumentar con el estándar abierto para no quedar atado.
4. Declarar objetivos de nivel de servicio y alertar por ritmo de consumo.
5. Correlacionar desde la alerta hasta la causa sin eslabones rotos.

Conceptos centrales

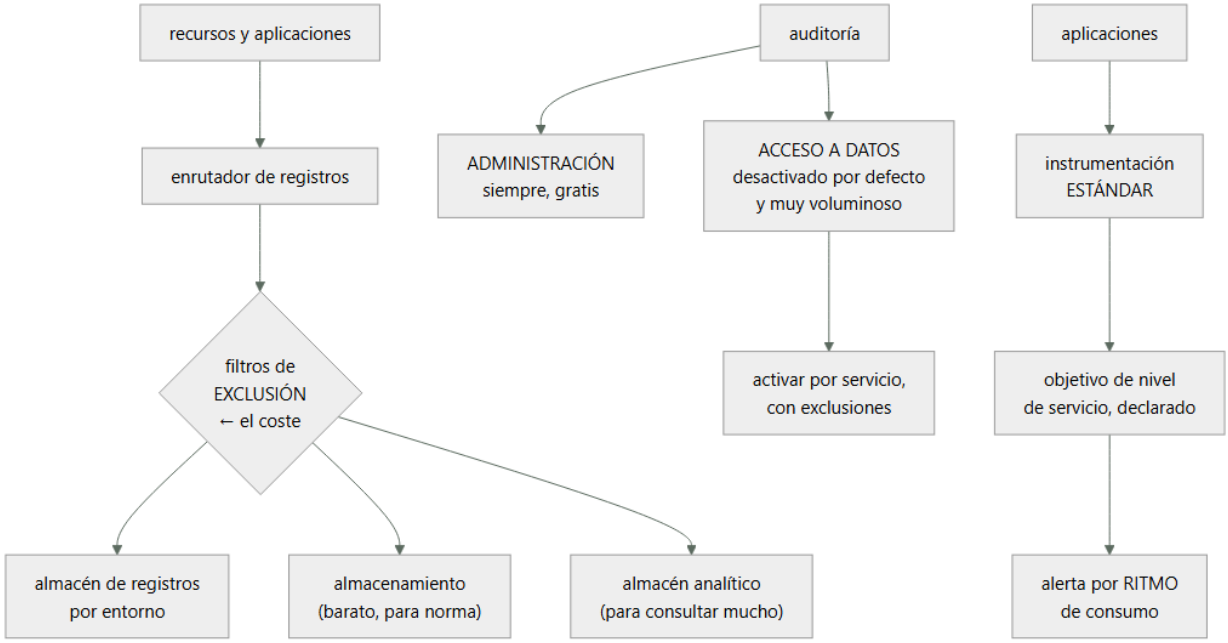
Concepto	Comprensión verificable
receptor de registros	Regla que enruta registros a un destino: almacén de registros, almacenamiento, almacén analítico o mensajería.
filtro de exclusión	Regla que descarta registros antes de almacenarlos. La palanca de coste más directa.
registro de auditoría de administración	Quién creó, modificó o borró qué. Siempre activo y sin coste.
registro de acceso a datos	Quién leyó o escribió datos. Desactivado por defecto y muy voluminoso.
objetivo de nivel de servicio	Recurso declarado con su indicador, su objetivo y su presupuesto de error.
alerta por ritmo de consumo	Se dispara cuando el presupuesto de error se gasta demasiado rápido. La que debe despertar.

Modelo mental

Google Cloud combina una jerarquía estricta de recursos con una red global y servicios data-first; proyecto, cuota e IAM forman una unidad operativa.

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. Enrutar y excluir

El modelo aquí es un enrutador: todo pasa por él y se decide qué va a dónde.

```
LOS DESTINOS
  almacén de registros   consultable, con retención
  almacenamiento         barato, para conservar por norma
  almacén analítico      para consultar mucho volumen
                                          clase 236
  mensajería             para procesar en tiempo real

Y EL RECEPTOR
  un filtro decide qué se envía a cada destino
  → y se pueden crear en la organización, para que apliquen
    a todos los proyectos                  clase 229
```

Los filtros de exclusión, que son la palanca de coste más directa:

```
se descartan ANTES de almacenar y no se pagan

lo que casi siempre se excluye
  comprobaciones de salud del balanceador
  peticiones correctas de servicios muy hablados, con
    muestreo parcial
  registros de nivel de información de componentes ruidosos
  y las lecturas de metadatos de los agentes

y con MUESTREO
  «excluye el 95 % de las peticiones con código 200»
  → conserva una muestra y descarta el resto
  → y hay que tenerlo en cuenta al contar    clase 225
```

Y el receptor a la organización, que resuelve un problema de gobierno:

```
un receptor en la organización envía los registros de
AUDITORÍA de todos los proyectos a un destino centralizado
  en un proyecto de seguridad
  al que los equipos NO tienen acceso de escritura ni de
  borrado
→ y eso es lo que hace la auditoría creíble  clase 141
→ sin ello, quien compromete un proyecto borra sus huellas
```

Y la retención, por destino:

```
almacén de registros de aplicación        30 días
almacén de auditoría                      400 días
```

```
almacenamiento para conservación          lo que exija la
                                          norma
→ y hay que fijarla: los valores por defecto no
  corresponden a nada                              ley 26
```

2. Los registros de auditoría

Esta es la pieza distintiva y la que hay que entender antes de activarla.

```
ADMINISTRACIÓN
  quién creó, modificó o borró un recurso
  SIEMPRE activo y sin coste
  → y es el que responde «¿quién cambió esto?»

ACCESO A DATOS
  quién leyó o escribió datos
  DESACTIVADO por defecto, salvo unos pocos    ley 26
  → y es el que responde «¿quién leyó esta tabla?»
  → es decir, el que hace falta para investigar una fuga

  y su volumen
    cada lectura de cada objeto genera una entrada
    → en un almacén con mucho tráfico, es enorme
    → activarlo en todos los servicios dispara la factura

EVENTOS DE SISTEMA
  acciones que hace la propia plataforma
  siempre activo

DENEGACIÓN DE POLÍTICA
  intentos rechazados por una política
  → muy útil y de poco volumen
```

Y la decisión que hay que tomar servicio por servicio:

```
¿ACTIVAR ACCESO A DATOS AQUÍ?
  sí   donde hay datos sensibles: almacenes con datos
       personales, conjuntos analíticos, gestor de secretos
  sí   en el gestor de claves, siempre
  no   en almacenes de artefactos, registros y datos
       públicos

y dentro de los activados
  distinguir LECTURA de ESCRITURA
    la escritura es poco voluminosa y muy valiosa
    la lectura es lo que dispara el volumen
  → activar escritura en muchos, lectura en pocos

y excluir las identidades que generan ruido
  los propios agentes, los procesos de copia
  → con una exclusión, no desactivando el servicio entero
```

Y la comprobación de que sirve:

```
«¿quién leyó la tabla de clientes en las últimas 24 h?»
→ si no se puede contestar, el registro de acceso a datos
  no está donde hace falta
→ y esa pregunta llega el día de un incidente, no antes
                                                clase 226
```

Y el acceso a los propios registros:

```
quien puede leer los registros de auditoría ve mucho
  → acceso restringido y auditado
  → y quien puede BORRARLOS o cambiar el enrutado, menos
    aún
  → con política de organización que lo impida  clase 229
```

3. Instrumentar y declarar objetivos

La instrumentación, con la decisión de la clase 225:

```
ESTÁNDAR ABIERTO
  trazas, métricas y registros con la API común
  el destino es configuración
  → cambiar de proveedor de análisis no toca el código
```

```
  → y la misma instrumentación vale en las tres nubes
                                                clase 158

y aquí hay un detalle a favor
  los servicios gestionados ya emiten trazas y métricas
  → y el contexto se propaga si la aplicación lo respeta
  → una petición que cruza balanceador, servicio de
    contenedores y base aparece como una sola traza
```

Y lo que hay que cuidar, que es lo de siempre:

```
el contexto de traza viaja también por la mensajería
  → dentro del mensaje                          clase 237
dimensiones de baja cardinalidad              clase 211
registros estructurados con traza, servicio, versión y
  entorno
y sin secretos ni datos personales completos
```

Los objetivos de nivel de servicio, que aquí son un recurso declarado:

```
se declara
  el SERVICIO
  el INDICADOR: disponibilidad o latencia, con su fuente
  el OBJETIVO: 99,5 % en 28 días
  → y la plataforma calcula el presupuesto de error y su
    consumo

y se puede declarar como código, junto al servicio
                                                clase 232

Y LA ALERTA QUE IMPORTA
  por RITMO DE CONSUMO del presupuesto
  «se está gastando 14 veces más rápido de lo normal»
  → esa es la que despierta                     clase 211
  → y no «hubo un error»

  con dos ventanas
    corta, para lo que arde
    larga, para la degradación lenta
```

Y la disciplina que este programa exige:

```
el objetivo se define por FLUJO DE USUARIO, no por
componente                                     clase 123
  «el 99,5 % de las confirmaciones de pedido en menos de
   800 ms, medidas en el borde»
→ y medido donde el usuario lo sufre          clase 126
```

Y las alertas que faltan siempre:

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

POR ANTIGÜEDAD
  mensajes sin confirmar, temas de fallidos, sincronización
  certificados, retraso de replicación
                                    clases 237, 234, 235

Y POR NEGOCIO
  «pedidos confirmados por minuto por debajo del mínimo»
  → detecta lo que ninguna métrica técnica ve      ley 15
```

4. Consultar, correlacionar y pagar

Las consultas preparadas, que ahorran minutos durante un incidente:

```
«todo lo de esta traza»
«errores de este servicio en la última hora, por tipo»
«qué cambió en los recursos en la última hora»
  → del registro de auditoría de administración
«quién accedió a estos datos»
  → del registro de acceso a datos
«peticiones más lentas y qué tramo domina»
```

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

La cadena de diagnóstico, con el eslabón que aquí es directo:

```
alerta por ritmo de consumo del presupuesto
  → panel del servicio
    → trazas del periodo
      → registros de esas trazas
        → REGISTRO DE AUDITORÍA: ¿qué se cambió?

→ y ese último eslabón es el que suele faltar en otras
  plataformas y aquí está siempre activo y gratis
```

Y el perfilado, que resuelve una categoría de problemas:

```
el perfilador continuo muestra dónde se va el tiempo y la
memoria en producción, con muestreo y coste bajo
→ y resuelve las preguntas que ni las trazas ni los
  registros contestan: «¿por qué esta función consume el
  40 % de la CPU?»
→ conviene tenerlo activo antes de necesitarlo
```

El coste, con las palancas:

```
1  FILTROS DE EXCLUSIÓN          ← la mayor
2  activar acceso a datos solo donde hace falta
3  retención por almacén, no global
4  destino según uso: consultable, barato o analítico
5  muestreo en la aplicación
6  y dimensiones de baja cardinalidad

y la revisión trimestral
  ¿cuánto cuesta la observabilidad frente al cómputo?
  → en la clase 211 costaba el doble
  → y aquí, con acceso a datos mal acotado, puede costar
    más                                           clase 239
```

Y la lista de comprobación de la clase:

- ■ hay receptor en la organización para los registros de auditoría
- ■ el destino de auditoría está en un proyecto sin acceso de escritura desde producción
- ■ hay filtros de exclusión para lo ruidoso
- ■ el acceso a datos está activado donde hay datos sensibles, y solo ahí
- ■ se distingue lectura de escritura al activarlo
- ■ la retención está fijada por almacén
- ■ el acceso a los registros está restringido y auditado
- ■ una política impide cambiar el enrutado de auditoría
- ■ la instrumentación de las aplicaciones es estándar
- ■ el contexto de traza viaja por la mensajería
- ■ hay objetivos declarados por flujo de usuario
- ■ hay alerta por ritmo de consumo, con dos ventanas
- ■ hay alertas por ausencia, por antigüedad y de negocio
- ■ las consultas de diagnóstico están guardadas
- ■ el perfilador continuo está activo
- ■ se puede contestar quién leyó unos datos concretos

Y el cierre que enlaza con la clase siguiente: con la observabilidad en pie, quedan la seguridad operativa y el coste, que aquí tienen herramientas propias y un mecanismo de perímetro que ya ha aparecido varias veces. Es la materia de la clase 239.

Ejemplo trabajado

CloudShop monta la observabilidad de su organización en Google Cloud. Lo que sigue es la activación del acceso a datos que multiplicó la factura por seis, la pregunta que no se podía contestar durante un incidente, y los objetivos declarados que sustituyeron a 71 alertas técnicas.

El punto de partida:

```
coste mensual de observabilidad            1.100 €
  registros                                  740 €
```

```
  métricas                                    240 €
  trazas                                      120 €

registros de auditoría
  de administración                        activos (gratis)
  DE ACCESO A DATOS                        desactivados
  destino                        el proyecto de cada equipo
  → cada equipo podía borrar sus propios registros de
    auditoría                                 clase 141
```

El incidente que reveló el hueco.

```
seguridad recibió un aviso: un correo de un cliente
aparecía en una lista de contactos de un tercero

la pregunta
  «¿quién ha leído la tabla de clientes en los últimos 90
   días?»

la respuesta
  no se podía contestar
  el registro de acceso a datos estaba desactivado
  el de administración decía quién había concedido permisos,
  no quién había leído

lo que se pudo hacer
  reconstruir con los registros de consulta del almacén
  analítico, que sí guardaban las consultas
  → 41 identidades habían consultado la tabla
  → y de ellas, 9 no debían tener acceso   clase 236

tiempo de investigación                        3 días
y la conclusión
  la respuesta se obtuvo por casualidad, porque el almacén
  analítico registra sus consultas
  → sobre el almacén de objetos no habría habido nada
```

La activación, y la factura.

```
primer intento
  se activó el acceso a datos en TODOS los servicios, con
  lectura y escritura

  factura del mes siguiente
    registros                          740 € → 6.400 €

  desglose
    almacén de objetos, lecturas            4.100 €
      → cada descarga de una imagen del catálogo generaba
        una entrada; 41 M al día
    almacén analítico, lecturas               980 €
    bases de datos, lecturas                  710 €
    resto                                     610 €
```

Y la corrección, servicio por servicio:

```
servicio                lectura   escritura   motivo
almacén con datos de
  clientes                 SÍ         SÍ      datos
                                              personales
almacén de imágenes
  del catálogo             no         SÍ      público
almacén analítico          SÍ         SÍ      datos
                                              personales
gestor de secretos         SÍ         SÍ      siempre
gestor de claves           SÍ         SÍ      siempre
base de pedidos            no         SÍ      volumen; y
                                              el acceso va
                                              por la API
almacén de artefactos      no         SÍ      no sensible
registros                  no         SÍ

y las exclusiones dentro de los activados
  identidades de los agentes de copia
  identidad del proceso de replicación
```

```
  → 61 % del volumen restante, excluido

registros                    6.400 € → 1.180 €
```

Y la comprobación:

```
«¿quién ha leído la tabla de clientes en las últimas 24 h?»
→ contestada en 40 segundos
→ y esa consulta quedó guardada y enlazada desde el
  procedimiento de incidente de datos          clase 127
```

El destino centralizado:

```
receptor en la ORGANIZACIÓN
  filtro: todos los registros de auditoría
  destino: proyecto de seguridad
    los equipos no tienen escritura ni borrado ahí
  retención                                   400 días
  y a partir de 90 días, a almacenamiento barato

y una política de organización
  nadie puede modificar ni borrar los receptores de
  auditoría de la organización              clase 229

prueba negativa
  intentar borrar el receptor desde un proyecto de
  producción                                  → denegado
  intentar borrar registros del proyecto de seguridad
                                              → denegado
                                                    ley 22
```

Los filtros de exclusión:

```
volumen antes                             1,9 TB/mes

exclusiones
  comprobaciones de salud del balanceador       -410 GB
  peticiones 200 del servicio de catálogo,
    muestreo del 3 %                            -620 GB
  lecturas de metadatos de los agentes          -180 GB
  registros de nivel de información de 3
    componentes ruidosos                        -240 GB

volumen después                            450 GB/mes
```

Los objetivos, que sustituyeron a las alertas técnicas:

```
antes
  alertas configuradas                              98
    disparadas al mes                              740
    accionables                                 61 (8 %)
  la mayoría, umbrales técnicos: CPU, memoria, errores

después
  objetivos declarados                               6
    confirmación de pedido: 99,5 % < 800 ms
    listado de catálogo: 99,9 % < 400 ms
    búsqueda: 99 % < 1 s
    procesamiento de eventos: 99,5 % < 5 min de retraso
    disponibilidad de la API pública: 99,8 %
    y el flujo de pago: 99,9 %

  alertas por ritmo de consumo                       6
    ventana corta (1 h) y larga (6 h)
  alertas por ausencia                              14
  alertas por antigüedad                             9
  alertas de negocio                                 4
  técnicas que sobrevivieron                        12
  ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  total                                             45

  disparadas al mes                                 68
  accionables                                   61 (90 %)

y las 71 técnicas retiradas
```

```
  → ninguna había detectado un problema que los objetivos
    no detectaran antes
```

La prueba de las alertas:

```
condición provocada en las 45
  no llegaron                                          7
    4 por umbral mal puesto
    2 por destino sin destinatarios
    1 porque el objetivo estaba mal declarado: el
      indicador medía en el servidor, no en el borde
                                       clase 126, ley 22
```

El perfilador, que resolvió algo que las trazas no:

```
síntoma   el servicio de búsqueda consumía el doble de CPU
          que hacía tres meses, con el mismo tráfico
          las trazas no mostraban ningún tramo lento

el perfilador continuo
  el 41 % de la CPU en una función de normalización de
  texto
  → un cambio de tres meses antes había añadido una
    expresión regular compilada EN CADA LLAMADA

corrección    compilar una vez
CPU                                          -38 %
instancias necesarias                         9 → 6
```

El resultado:

```
coste de observabilidad                   1.100 €     1.640 €
  (con auditoría de datos activada donde hace falta)
  frente a 6.400 € con todo activado
volumen de registros                       1,9 TB      450 GB
«¿quién leyó estos datos?»            incontestable    40 s
registros de auditoría borrables por
  los equipos                                  sí          no
alertas configuradas                           98          45
  accionables                                61 (8 %)   61 (90 %)
objetivos declarados                              0           6
alertas que no llegaban                         n/d      0 (de 7
                                                       corregidas)
```

La lección que esta clase deja: activar el registro de acceso a datos en todos los servicios multiplicó la factura por seis, y acotarlo a donde hay datos sensibles la dejó en menos de la cuarta parte sin perder la capacidad que importaba. La pregunta que un incidente hizo —quién leyó una tabla— no se podía contestar, y solo se respondió por casualidad porque el almacén analítico registra sus consultas. Y setenta y una alertas técnicas se retiraron sin perder nada: ninguna había detectado un problema antes que los objetivos.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-19-gcp-production-architecture/238-cloud-operations-trace-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 gcp-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 gcp-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
No se puede saber quién leyó unos datos durante un incidente	El registro de acceso a datos está desactivado por defecto	Actívalo en los servicios con datos sensibles, distinguiendo lectura de escritura, y guarda la consulta que responde esa pregunta.
La factura de registros se multiplica al activar la auditoría de datos	Se activó la lectura en todos los servicios, incluidos los de mucho volumen y datos públicos	Acota por servicio, activa escritura en muchos y lectura en pocos, y excluye las identidades de agentes y procesos de copia.
Un equipo puede borrar los registros que lo incriminan	La auditoría se guarda en el proyecto del propio equipo	Receptor en la organización hacia un proyecto de seguridad sin escritura desde producción, y política que impida modificarlo.
Hay decenas de alertas y ninguna dice si el usuario está sufriendo	Son umbrales técnicos y no objetivos por flujo de usuario	Declara objetivos por flujo, alerta por ritmo de consumo del presupuesto y retira las técnicas que no aporten.
Un objetivo declarado no detecta un problema real	El indicador se mide en el servidor y no donde el usuario lo sufre	Mide en el borde y comprueba el objetivo provocando la condición.
Un servicio consume más CPU y las trazas no muestran nada	El coste está dentro del proceso, no en las llamadas	Activa el perfilador continuo antes de necesitarlo; muestra dónde se va el tiempo en producción.

■ 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 registro de administración y el de acceso a datos?
2. ¿Por qué el destino de la auditoría debe estar fuera del alcance de producción?
3. ¿Cuál es la palanca de coste más directa en el enrutado de registros?
4. ¿Qué alerta debe despertar a alguien y por qué no el error suelto?

5. ¿Qué resuelve el perfilador continuo que no resuelven trazas ni registros?

Referencias

- Google Cloud (2025). Cloud Logging: routing and storage. <https://cloud.google.com/logging/docs/routing/overview>
- Google Cloud (2025). Cloud Audit Logs. <https://cloud.google.com/logging/docs/audit>
- Google Cloud (2025). Service monitoring: SLOs and burn rate alerts. <https://cloud.google.com/stackdriver/docs/solutions/slo-monitoring>
- Google Cloud (2025). Cloud Trace and OpenTelemetry. <https://cloud.google.com/trace/docs/setup>
- Google Cloud (2025). Cloud Profiler. <https://cloud.google.com/profiler/docs/about-profiler>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 19 en PDF · Recorrido de Google Cloud en PDF · Manual integral

Anterior	Índice	Siguiente
← 237 · Pub/Sub, Eventarc y entrega exactamente-una-vez	Parte 19 · Programa	239 · SCC, VPC Service Controls, KMS y FinOps →

Evaluación — 238 Cloud Operations, Trace 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 gcp-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.

239 — SCC, VPC Service Controls, KMS y FinOps

Parte: 19 — Google Cloud: arquitectura de datos y operación en producción Nivel: avanzado · Horas estimadas: 4 Laboratorio: security · Estado: EXECUTABLECORE

Propósito

Cerrar la operación de Google Cloud con la seguridad y el coste, que aquí tienen dos piezas propias: el perímetro de servicio, que es el control que impide sacar datos aunque los permisos lo permitan, y el modelo de descuentos automáticos, que cambia el cálculo de los compromisos. La clase cubre la postura y la detección con la disciplina de la clase 226, y el control de coste con la de la 214.

Resultados de aprendizaje

Al finalizar podrás:

1. Priorizar los hallazgos de seguridad por alcance y por camino de ataque.
2. Aplicar perímetros de servicio y entender qué cierran.
3. Gestionar claves de cifrado propias donde aportan.
4. Atribuir el coste y aprovechar los descuentos automáticos.
5. Comprometer capacidad solo tras retirar y redimensionar.

Conceptos centrales

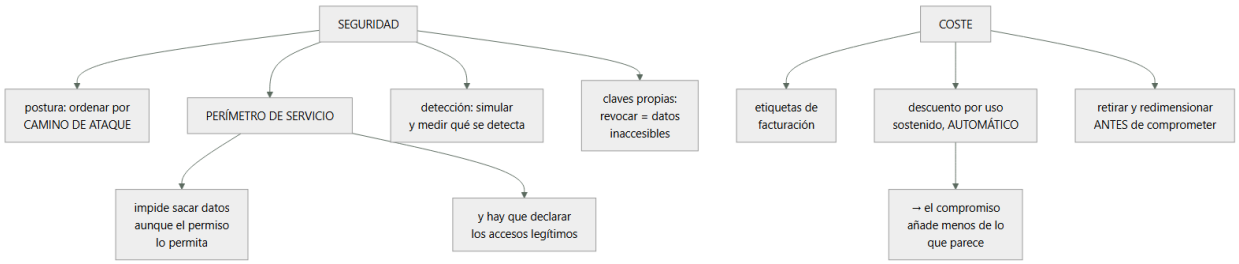
Concepto	Comprensión verificable
perímetro de servicio	Frontera que restringe qué API pueden usarse hacia dentro y hacia fuera de un grupo de proyectos.
camino de ataque	Secuencia de pasos desde un punto expuesto hasta un dato valioso. Ordena la prioridad.
clave gestionada por el cliente	Clave de cifrado bajo control propio. Permite revocar el acceso a los datos por completo.
descuento por uso sostenido	Rebaja automática por mantener recursos encendidos, sin compromiso.
compromiso de uso	Descuento a cambio de un compromiso de gasto o de capacidad durante uno o tres años.
etiqueta de facturación	Etiqueta que aparece en el desglose de costes. Distinta de las que se usan en condiciones de permiso.

Modelo mental

Google Cloud combina una jerarquía estricta de recursos con una red global y servicios data-first; proyecto, cuota e IAM forman una unidad operativa.

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 y caminos de ataque

La disciplina es la de la clase 226 y aquí hay una herramienta que la hace directa.

```
LO QUE PRODUCE LA EVALUACIÓN CONTINUA
  hallazgos de configuración: público, sin cifrar, sin
  copias, versión antigua
  hallazgos de identidad: permisos amplios, claves
  hallazgos de vulnerabilidades

Y LO QUE ORDENA EL TRABAJO
  no la gravedad declarada
  sino los CAMINOS DE ATAQUE
  «desde esta máquina expuesta, con esta cuenta de
   servicio, se llega a este conjunto con datos de
   clientes»
  → con la puntuación de exposición calculada

→ y esos caminos concentran más riesgo que cientos de
  hallazgos sueltos                          clase 226
```

Y el orden de trabajo, ya conocido:

```
1  cerrar los caminos que terminan en datos
2  reducir el alcance de las identidades más amplias
                                              clase 230
3  impedir por política que lo arreglado vuelva
                                              clase 229
4  remediar el resto por lotes                  ley 27
5  aceptar por escrito lo que no compensa     clase 189
```

Y los hallazgos que se repiten en esta nube:

```
cuentas de servicio con papel de proyecto     clase 230
claves de cuenta de servicio sin caducidad
almacenes con acceso a «todos los usuarios autenticados»
  → que significa cualquier cuenta de Google, no solo la
    organización
  → y es el hallazgo que más sorprende
reglas de cortafuegos con selector por etiqueta clase 229
conjuntos analíticos con acceso al conjunto entero
                                              clase 236
y proyectos sin carpeta, que no reciben políticas
```

Y la comprobación de la detección:

```
simular técnicas conocidas y medir qué proporción se
detecta
→ es la única medida real                   clase 226
→ y las que no se detecten son la lista de trabajo
```

2. El perímetro de servicio

Este es el control propio de esta nube y el que cierra el hueco que dejan los permisos.

```
EL PROBLEMA QUE RESUELVE
  una identidad con permiso legítimo sobre unos datos puede
  copiarlos a un proyecto de otra organización
  → el permiso lo autoriza
  → la red no lo impide, porque va por la API de la
    plataforma                              clase 200, 231
```

```
EL PERÍMETRO
  agrupa proyectos y restringe qué API pueden usarse
  desde fuera hacia dentro     ← quién puede leer
  desde dentro hacia fuera     ← a dónde se puede escribir
  → y una identidad de dentro NO puede escribir en un
    recurso de fuera del perímetro
```

Y lo que hay que hacer para que funcione sin romper nada:

```
1  DEFINIR el perímetro: qué proyectos y qué servicios
2  MODO DE SIMULACIÓN: registra lo que bloquearía
3  MEDIR semanas
4  DECLARAR los accesos legítimos que cruzan
     reglas de entrada y de salida, por identidad, por
     proyecto y por servicio
5  APLICAR

→ activar sin simular corta integraciones legítimas
  → y es la misma disciplina de las clases 200, 209 y 217
```

Y lo que suele aparecer en la simulación:

```
herramientas de terceros que leen datos
canalizaciones de otro proyecto
copias hacia un proyecto de análisis
el propio equipo consultando desde su portátil
  → y eso obliga a decidir: ¿se declara o se prohíbe?
```

Las claves gestionadas por el cliente, con lo que aportan de verdad:

```
QUÉ APORTAN
  el control de la clave: revocarla hace los datos
  inaccesibles, incluso para el proveedor
  → y eso es lo que exigen algunos requisitos
                                                clase 177
  rotación bajo control propio
  y registro de cada uso de la clave           clase 238

QUÉ CUESTAN
  operación: rotación, permisos, disponibilidad de la clave
  y si la clave se pierde o se revoca por error, los datos
  son irrecuperables
  → esa es una consecuencia real, no teórica

DÓNDE APLICARLAS
  donde haya requisito o datos sensibles
  no en todo                                    ley 26
```

Y una advertencia operativa:

```
la clave vive en una región
→ si esa región no está disponible, los datos cifrados con
  ella tampoco
→ y eso entra en el cálculo del techo de disponibilidad
                                                clase 185
```

3. Coste: los descuentos que ya se aplican

Aquí hay una diferencia que cambia el cálculo de los compromisos.

```
DESCUENTO POR USO SOSTENIDO
  se aplica AUTOMÁTICAMENTE por mantener recursos
  encendidos durante el mes
  → sin compromiso, sin contratar nada
  → y llega a ser apreciable

Y LA CONSECUENCIA
  el compromiso de uso añade descuento SOBRE lo ya
  descontado
  → el ahorro incremental es menor de lo que sugiere el
    porcentaje anunciado
  → y hay que calcularlo con las cifras propias, no con el
    porcentaje del catálogo              clase 227
```

Y los tipos de compromiso:

```
POR GASTO
```

```
  se compromete una cantidad por hora
  + flexible: se aplica a lo que haya
  − descuento menor

POR RECURSO
  se compromete una cantidad de CPU y memoria en una región
  + descuento mayor
  − rígido: si se cambia de región o de familia, se pierde

→ y el orden de la clase 227 sigue valiendo
  retirar → redimensionar → apagar → y ENTONCES comprometer
```

La atribución, con la particularidad de las etiquetas:

```
HAY DOS MECANISMOS DE ETIQUETADO
  uno aparece en el desglose de FACTURACIÓN
  otro sirve para CONDICIONES de permiso y para políticas
                                          clases 229, 230
  → y no son intercambiables
  → usar el equivocado hace que el gasto no se atribuya

Y LA JERARQUÍA AYUDA
  el desglose por PROYECTO es directo y limpio
  → y por eso la separación por proyecto de la clase 229
    es también una decisión de atribución
```

Y el resto del método, que es el de la clase 214:

```
etiquetas obligatorias en la creación, por política
presupuestos por proyecto, sobre previsión, con acción en
  no producción
detección de anomalías por servicio y por dueño
retirada automática de lo ocioso: apagar antes de borrar
y coste por unidad de negocio
```

Y las partidas que sorprenden en esta nube:

```
consultas del almacén analítico              clase 236
registros de acceso a datos                  clase 238
almacenamiento de mensajes sin confirmar     clase 237
tráfico entre zonas y entre regiones
mínimos de capacidad de las bases distribuidas clase 235
y los proyectos sin uso que nadie retira       ley 25
```

4. El ritmo, y lo que se degrada solo

Coste y seguridad comparten la propiedad de degradarse sin intervención, y esta clase cierra la parte con el mismo calendario de la 227.

```
SEMANAL
  anomalías de coste abiertas
  hallazgos críticos nuevos
  recursos ociosos retirados

MENSUAL
  coste por unidad de negocio y su tendencia
  cobertura y utilización de compromisos
  proyectos sin uso
  y hallazgos por camino de ataque

TRIMESTRAL
  simulación de técnicas y proporción detectada clase 226
  prueba del acceso de emergencia            clase 230
  revisión de accesos y de exenciones
  y un experimento de resiliencia            clase 227

ANUAL
  ejercicio de pérdida de región             clase 215
  revisión de las decisiones registradas     clase 190
```

Y las señales que dicen si está sano:

```
SEGURIDAD
  caminos de ataque que terminan en datos      → cero
  proporción de técnicas simuladas detectadas
  claves de cuenta de servicio                 → cero
```

```
  proyectos sin carpeta                          → cero
  y perímetros con reglas de excepción sin fecha → cero

COSTE
  gasto atribuido                                > 90 %
  coste por unidad de negocio                    tendencia
  utilización de compromisos                     > 95 %
  proyectos sin uso                              tendencia a
                                                 cero
```

Y una comprobación honesta, la misma de la clase 226:

```
¿cuántos incidentes detectó el centro y cuántos se
detectaron por otra vía?
→ y las vías de siempre: una factura, un tercero, un
  inventario, una persona
```

Y la lista de comprobación de la clase:

- los hallazgos se ordenan por camino de ataque
- no hay caminos que terminen en datos sensibles
- no hay almacenes con acceso a todos los usuarios autenticados
- hay perímetros de servicio sobre los proyectos con datos
- los perímetros se simularon antes de aplicar
- las reglas de excepción del perímetro tienen dueño y fecha
- las claves propias están donde hay requisito, y su región entra en el cálculo de disponibilidad
- las etiquetas de facturación son las correctas
- el gasto atribuido supera el 90 %
- se retiró y redimensionó antes de comprometer
- el descuento incremental del compromiso se calculó sobre el ya aplicado
- hay presupuestos con acción y detección de anomalías
- se simulan técnicas y se publica la proporción detectada
- existe el 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, compararlo con las otras dos nubes y cerrar la parte. Es la materia de la clase 240.

Ejemplo trabajado

CloudShop cierra la operación de Google Cloud. Lo que sigue son los almacenes accesibles por cualquier cuenta de Google, el perímetro que cortó una integración legítima, y el cálculo de compromiso que resultó dar mucho menos de lo anunciado.

La postura, ordenada por camino de ataque:

```
hallazgos abiertos                                  2.140
puntuación de exposición: los 5 caminos principales

1  máquina con IP externa y puerto de administración
   abierto
   → su cuenta de servicio tiene papel de proyecto
     → alcanza el almacén de copias y la base de pedidos
   pasos: 3 · datos al final: 2,3 M de registros

2  almacén de exportaciones con acceso a «todos los
   usuarios autenticados»
   → cualquier cuenta de Google del mundo podía listarlo y
     descargarlo
   → contenía extractos diarios de pedidos desde 2023
   pasos: 1 · datos: 41 meses de extractos

3  cuenta de servicio de la canalización con papel de
   organización                                clase 230

4  conjunto analítico con acceso al conjunto entero para
   41 personas                                 clase 236

5  clave de cuenta de servicio de un consultor externo,
   activa desde 2023                            ley 25
```

Y el segundo merece detalle:

```
«todos los usuarios autenticados» NO significa «todos los
de la organización»
  significa cualquiera con una cuenta de Google

→ el equipo lo había configurado creyendo lo primero
→ y la política de organización que lo impide existía
  desde la clase 229, pero el almacén se había creado ANTES
                                                    ley 27

cuánto llevaba así                              19 meses
accesos externos registrados            no se sabía
  → el registro de acceso a datos no estaba activo en ese
    almacén                                     clase 238
  → se activó, y en 30 días: 0 accesos externos
  → lo cual no dice nada de los 19 meses anteriores
```

El perímetro de servicio, con su simulación.

```
definición
  proyectos: los 14 de producción con datos
  servicios restringidos: almacenamiento, almacén
  analítico, bases, gestor de secretos y claves

modo de simulación, 4 semanas
  accesos que se bloquearían                    41.200
    legítimos                                      610
    del propio perímetro                        40.400
    NO IDENTIFICADOS                               190  ←

los 610 legítimos
  · canalización de despliegue desde el proyecto de
    plataforma
  · herramienta de visualización de un tercero
  · exportación nocturna al proyecto de análisis
  · 4 personas consultando desde su portátil

los 190 no identificados
  · 140 de una cuenta de servicio de un proyecto que se
    creía retirado en 2024                      ley 25
  ·  38 de una herramienta de un proveedor cuyo contrato
    había terminado
  ·  12 de una dirección que resultó ser un entorno de
    pruebas de un socio, no declarado

→ los 190 se cortaron a propósito
→ y los 610 se declararon como reglas de entrada y salida,
  con dueño y fecha de revisión
```

Y el corte que ocurrió igualmente:

```
al aplicar, una integración se rompió
  el equipo de finanzas exportaba un informe mensual desde
  una hoja de cálculo conectada
  → esa conexión no había aparecido en las 4 semanas de
    simulación porque es MENSUAL y cayó fuera de la
    ventana

→ duración del corte                             2 días
→ corrección: regla declarada
→ y lección: la ventana de simulación debe cubrir el ciclo
  completo de negocio                          clase 167
```

Las claves propias, decididas por caso:

```
dónde se aplicaron
  almacén con datos de clientes
  conjunto analítico con datos personales
  copias de seguridad
  → los tres con requisito contractual del mayor cliente

dónde NO
  almacenes de imágenes y artefactos
  registros de aplicación
```

```
  → sin requisito y con coste de operación

y el efecto sobre disponibilidad, calculado
  la clave vive en una región
  → se configuró con réplica en la segunda región
  → sin eso, el techo del flujo habría bajado  clase 185

y la prueba negativa
  revocar el acceso a la clave en preproducción
  → los datos pasaron a ser inaccesibles, correctamente
  → y la restauración tardó 11 minutos
  → procedimiento escrito y probado               ley 22
```

El cálculo de compromiso, que dio menos de lo anunciado.

```
gasto de cómputo                            8.400 €/mes

el descuento por uso sostenido YA aplicado
  precio de lista                          11.900 €
  descuento automático                     -3.500 €
  gasto real                                8.400 €

la oferta de compromiso a 1 año
  descuento anunciado sobre lista              37 %
  → parecía un ahorro de 4.400 €/mes

el cálculo real
  con compromiso, sobre lista               7.500 €
  ahorro frente a los 8.400 actuales          900 €/mes
  → no 4.400 €

→ porque el descuento automático ya estaba aplicado
→ y comparar con el precio de lista es el error
                                                  clase 227
```

Y el orden que se siguió antes de comprometer:

```
1  RETIRAR   71 proyectos sin actividad      clase 229
             -3.100 €/mes
2  REDIMENSIONAR  peticiones de recursos del clúster
                                             clase 234
             -1.760 €/mes
3  APAGAR    entornos no productivos por horario
             -1.900 €/mes
4  y ENTONCES comprometer, sobre la base estable
             -610 €/mes

→ y el compromiso se hizo POR GASTO, no por recurso
→ motivo: el clúster iba a cambiar de familia
```

El coste, atribuido:

```
al empezar
  gasto atribuido                              64 %
  y el 36 % restante
    recursos con el tipo de etiqueta EQUIVOCADO   2.100 €
      → se habían puesto las de condiciones de permiso,
        que no aparecen en el desglose
    servicios compartidos                      1.900 €
    tráfico entre zonas                        1.400 €

tras corregir el tipo de etiqueta y repartir lo compartido
  gasto atribuido                              95 %
```

La detección, medida:

```
simulación de 14 técnicas
  detectadas la primera vez                  9 de 14
  las 5 que no
    · exfiltración al almacén de otra organización
      → la detectó el PERÍMETRO, no la regla
      → y se aceptó así
    · suplantación de una cuenta privilegiada
      → regla escrita
    · creación de una clave de cuenta de servicio
```

```
      → la impide la política; regla añadida igualmente
    · acceso a datos desde una identidad inusual
      → dependía del registro de acceso a datos, ahora
        activo
    · enumeración de proyectos
      → umbral corregido

segunda simulación                          13 de 14
```

El resultado:

```
caminos de ataque que terminan en datos         5           0
almacenes accesibles por cualquier cuenta       3           0
perímetros de servicio                          0           2
accesos no identificados que cruzaban         190           0
claves de cuenta de servicio                   12           3
gasto atribuido                               64 %        95 %
coste mensual                              21.400 €    14.030 €
utilización del compromiso                        —        97 %
técnicas simuladas detectadas                  9/14       13/14
```

La lección que esta clase deja: un almacén con extractos de pedidos de cuarenta y un meses era descargable por cualquier persona con una cuenta de Google, porque «todos los usuarios autenticados» no significa lo que parece, y la política que lo impedía se creó después que el almacén. Y el compromiso que prometía un 37 % de descuento daba novecientos euros al mes y no cuatro mil cuatrocientos, porque el descuento automático ya estaba aplicado: comparar con el precio de lista es el error.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-19-gcp-production-architecture/239-scc-vpc-service-controls-kms-y-finops/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 gcp-security-finops 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 gcp-security-finops 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 almacén resulta accesible desde fuera de la organización	Se concedió a «todos los usuarios autenticados», que significa cualquier cuenta del proveedor	Prohíbelo por política, revisa lo existente porque la política no actúa hacia atrás, y activa el registro de acceso a datos allí.
Una identidad legítima copia datos a otra organización	El permiso lo autoriza y la red no lo impide porque va por la API	Aplica perímetros de servicio sobre los proyectos con datos, simulando antes y declarando los accesos legítimos.
Aplicar el perímetro corta una integración que no apareció en la simulación	La ventana de simulación no cubrió un proceso mensual	Simula durante un ciclo completo de negocio antes de aplicar.
El ahorro del compromiso es mucho menor de lo anunciado	Se comparó con el precio de lista y el descuento automático ya estaba aplicado	Calcula el ahorro incremental sobre el gasto real actual, no sobre la tarifa de catálogo.
Parte del gasto no aparece atribuido pese a etiquetar	Se usó el tipo de etiqueta que sirve para condiciones de permiso y no el de facturación	Comprueba qué mecanismo aparece en el desglose y usa ese; los dos no son intercambiables.
Los datos cifrados con clave propia dejan de estar disponibles	La clave vive en una región que no está disponible	Replica la clave y cuenta su disponibilidad en el cálculo del techo del flujo.

Seguridad, ética y costo

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

Preguntas de comprobación

1. ¿Qué ordena la prioridad de los hallazgos de seguridad?
2. ¿Qué cierra un perímetro de servicio que no cierra ni el permiso ni el cortafuegos?
3. ¿Qué hay que comprobar antes de aplicar un perímetro y durante cuánto tiempo?
4. ¿Por qué el descuento por compromiso da menos de lo que anuncia el porcentaje?
5. ¿Qué efecto tiene sobre la disponibilidad usar claves de cifrado propias?

Referencias

- Google Cloud (2025). Security Command Center: attack path simulation. <https://cloud.google.com/security-command-center/docs/attack-exposure-learn>
- Google Cloud (2025). VPC Service Controls: dry run mode. <https://cloud.google.com/vpc-service-controls/docs/dry-run-mode>
- Google Cloud (2025). Customer-managed encryption keys. <https://cloud.google.com/kms/docs/cmek>
- Google Cloud (2025). Sustained use and committed use discounts. <https://cloud.google.com/compute/docs/sustained-use-discounts>
- Google Cloud (2025). Billing labels and cost breakdown. <https://cloud.google.com/billing/docs/how-to/bq-examples>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 19 en PDF · Recorrido de Google Cloud en PDF · Manual integral

Anterior	Índice	Siguiente
← 238 · Cloud Operations, Trace y OpenTelemetry	Parte 19 · Programa	240 · Proyecto: CloudShop productivo en Google Cloud →

Evaluación — 239 SCC, VPC Service Controls, KMS y FinOps

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 gcp-security-finops 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.

240 — Proyecto: CloudShop productivo en Google Cloud

Parte: 19 — Google Cloud: arquitectura de datos y operación en producción Nivel: avanzado · Horas estimadas: 8 Laboratorio: capstone · Estado: EXECUTABLECORE

Propósito

Poner en producción el sistema completo de CloudShop en Google Cloud, compararlo con las otras dos nubes y cerrar la parte 19. La clase da el orden, el entregable y las pruebas negativas; corrige las cinco predicciones de la clase 228 —tres acertadas, una incompleta y una fallada—, actualiza el recuento de leyes, añade la ley 28 y escribe la hipótesis de la parte 20.

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 las tres nubes con cifras y sacar conclusiones defendibles.
4. Corregir las cinco predicciones de la clase 228 con evidencia.
5. Escribir la hipótesis de la parte 20 en forma refutable.

Conceptos centrales

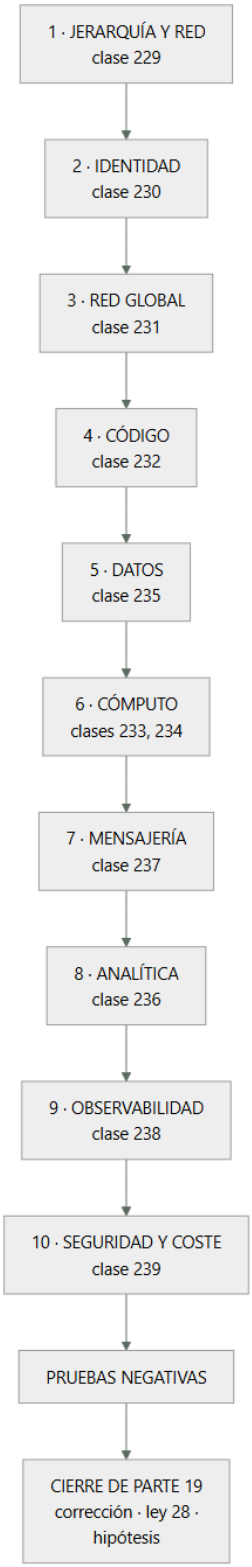
Concepto	Comprensión verificable
ley 28	Donde se paga por uso, cada valor por defecto es una factura; y nadie lo revisa porque no parece una decisión.
error de traslado	Fallo que viene de dar por válida una suposición de otra nube. La causa dominante de esta parte.
coste de operar varias nubes	Suma de modelos de gobierno, permisos, red, observabilidad y calendarios. No es proporcional al número.
equivalencia conceptual	Correspondencia de ideas entre proveedores. Alta; la operativa es la que difiere.
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

Google Cloud combina una jerarquía estricta de recursos con una red global y servicios data-first; proyecto, cuota e IAM forman una unidad operativa.

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 las pruebas

El encargo. Llevar a producción la plataforma de pedidos de CloudShop en Google Cloud, con usuarios reales, guardia, presupuesto y continuidad comprobada.

El orden, por coste de cambio:

```
1  JERARQUÍA Y RED COMPARTIDA                    clase 229
   carpetas por entorno, proyectos por carga
   políticas de organización en auditoría antes de aplicar
   → y con plan de remediación de lo existente    ley 27

2  IDENTIDAD                                     clase 230
   sin claves de cuenta de servicio
   asignaciones en el ámbito del recurso, con condiciones

3  RED GLOBAL                                    clase 231
   una red por entorno, salida cerrada, zonas privadas
   asociadas en el proyecto anfitrión

4  CÓDIGO                                        clase 232
   estado protegido, validación de políticas y coste antes
   de aplicar

5  DATOS                                         clase 235
   familia por patrones y por coste base; clave que reparte

6  CÓMPUTO                            clases 233, 234
   servicio de contenedores por defecto; clúster donde hace
   falta

7  MENSAJERÍA                                    clase 237
   un tema, una suscripción por consumidor, idempotencia

8  ANALÍTICA                                     clase 236
   particionado, límites de consulta y acceso por columna

9  OBSERVABILIDAD                                clase 238
   instrumentación estándar, objetivos declarados,
   auditoría acotada

10 SEGURIDAD Y COSTE                             clase 239
   perímetros, caminos de ataque, compromisos tras retirar
```

Las pruebas negativas de la parte:

- crear un recurso sin etiquetas obligatorias
- crear una clave de cuenta de servicio
- suplantar una cuenta sin permiso
- acceder fuera de una condición de recurso o de tiempo
- activar un papel privilegiado sin aprobación
- entrar con el acceso de emergencia
- resolver un nombre con conexión privada desde cada proyecto
- sacar datos a un proyecto de otra organización
- desplegar algo que incumple una política
- borrar el estado y restaurarlo
- ejecutar el plan sin cambios y comprobar que sale vacío
- escribir con clave secuencial y observar el reparto
- consultar una tabla grande sin filtro de partición
- consultar una columna sensible sin la etiqueta
- enviar el mismo mensaje 50 veces
- dejar un mensaje en el tema de fallidos y esperar alerta
- obtener la identidad desde otro espacio de nombres
- perder una región y cronometrar
- provocar la condición de cada alerta
- desplegar el entorno de cero en un proyecto vacío

Y los criterios de evaluación, con el peso donde importa:

```
1  ningún proyecto sin carpeta ni sin dueño                2
2  cero claves de cuenta de servicio sin excepción         3
3  el alcance de cada identidad está medido                3
4  las zonas privadas resuelven desde todos los proyectos  3
5  hay perímetro de servicio sobre los proyectos con datos 3
6  la clave de las tablas distribuidas reparte             2
7  hay límite de bytes por consulta                        3
8  el acceso a datos personales es por columna             3
9  la instrumentación es estándar                          2
10 hay objetivos declarados con alerta por ritmo           2
11 la lista de valores por defecto cambiados es explícita  3
```

```
12 las pruebas negativas se ejecutaron y hay fallos
   publicados                                          3
```

2. Cierre de la parte 19: corrección de las cinco predicciones

Las cinco predicciones de la clase 228, corregidas con la evidencia de las clases 229 a 239.

1. «la tercera nube confirmará la equivalencia conceptual y diferirá en lo operativo; y lo que más costará no será aprender lo distinto sino DESAPRENDER lo de las dos anteriores»

   CORRECTA, y con cinco casos documentados. Crear una red por proyecto; asumir que las reglas de cortafuegos son por subred; confundir la barrera con el permiso; poner la concurrencia en 1 por trasladar el modelo de funciones; y usar una clave autoincremental en un almacén repartido por rangos. El más puro fue el de las zonas privadas: el equipo HABÍA SUFRIDO ese problema en la nube anterior, lo conocía, y lo repitió porque el sitio donde se asocia la zona es distinto.

2. «la jerarquía de proyectos y la red compartida volverán a condicionar todo, y el error frecuente será el CONTRARIO: demasiados proyectos»

   CORRECTA y literal. 214 proyectos, 189 colgando de la organización sin carpeta, 71 sin actividad en 90 días, 120 sin dueño. Y uno de los inactivos contenía una copia de la tabla de clientes legible por cualquier cuenta, desde hacía 19 meses.

3. «la identidad volverá a ser el eje y el error más frecuente volverá a ser el ámbito amplio»

   CORRECTA E INCOMPLETA, y lo que falta es lo más útil. El ámbito amplio estaba: 18 asignaciones en la organización y 140 en carpetas. Pero el hallazgo mayor fue otro: eliminar 341 claves de cuenta de servicio costó cuatro meses y NO redujo el alcance ni un punto. Quitar credenciales y reducir alcance son dos trabajos distintos, y confundirlos hace creer que el primero resuelve el segundo. Esa distinción no estaba en la predicción.

4. «los valores por defecto volverán a estar mal elegidos, y aquí fallarán sobre todo en RED»

   FALLADA en la segunda mitad, y el patrón del fallo importa. Los valores por defecto de red estaban mal —la salida permitida, el selector por etiqueta—, pero los que costaron dinero de verdad fueron otros: consultar sin filtro de partición y con todas las columnas, 4.100 €/mes; activar la auditoría de acceso a datos en todo, ×6 la factura de registros; el indexado automático de documentos; la concurrencia; y una suscripción abandonada reteniendo 41 millones de mensajes. Es la SEGUNDA VEZ que fallamos en dónde fallarán los valores por defecto: en la clase 228 predijimos identidad para Azure y era coste y observabilidad; aquí predijimos red y era analítica y auditoría. El patrón que no habíamos visto es que los valores por defecto caros están donde se factura por uso, no donde la nube es característica.

5. «los problemas serán otra vez de las leyes 25, 15 y 22. Lo refutable: la proporción de hallazgos que detecte una comprobación automática y periódica será MAYOR que en las partes 17 y 18»

   CORRECTA, y medible. En esta parte, las comprobaciones automáticas detectaron: 4 proyectos con zonas sin asociar, 23 casos de deriva, 74 despliegues que incumplían política, 190 accesos no identificados que

```
   cruzaban el perímetro, 5 caminos de ataque y 5 técnicas
   no detectadas. Frente a las partes anteriores, donde casi
   todo lo grave lo encontraron inventarios o facturas.
   PERO el matiz cualitativo se mantiene: los tres hallazgos
   más graves de esta parte —el almacén público de 19 meses,
   la suscripción de 41 millones de mensajes y las 41
   personas con acceso a datos personales— los encontraron
   un inventario, una factura y otro inventario. Las
   comprobaciones detectan más cosas; las peores siguen
   apareciendo al mirar.
```

Marcador: tres correctas, una incompleta, una fallada.

3. Recuento de leyes, ley 28 e hipótesis de la parte 20

El recuento de leyes, cerrada la parte 19.

```
ley 13  lo que no se mira deja de funcionar en silencio        55
ley 15  la señal existe y nadie la mira                        44
ley 22  un procedimiento nunca ejecutado no funciona           39
ley 14  el coste se decide al crear, no al pagar               34
ley 16  un control que estorba se rodea                        32
ley 20  lo que no tiene dueño se filtra y se desperdicia       31
ley 21  el acoplamiento vive en quién escribe                  25
ley 25  lo provisional sobrevive a su motivo                   24
ley 26  el valor por defecto sirve a la demostración           19
ley 23  la capacidad la limita lo que ya se mantiene           19
ley 24  lo que no está en el diagrama no se analiza            13
ley 17  se optimiza la medida, no el objetivo                  13
ley 19  la compensación hace invisible el fallo                10
ley 27  un control solo actúa sobre lo que cambia               9
ley 18  lo asíncrono traslada la garantía, no la elimina        9
```

Y la parte 19 obliga a escribir la ley que explica el fallo de la predicción 4:

```
LEY 28
  donde se paga por uso, cada valor por defecto es una
  factura; y nadie lo revisa porque no parece una decisión

apariciones en esta parte                                      5
  clase 236   consultar sin filtro de partición y con todas
              las columnas: 4.100 €/mes en tres paneles
  clase 238   auditoría de acceso a datos activada en todo:
              la factura de registros ×6
  clase 237   una suscripción sin consumidor reteniendo
              41 M de mensajes: 1.520 €/mes de
              almacenamiento
  clase 235   indexado automático de todos los campos de un
              documento
  clase 233   concurrencia 1 por instancia: 2.900 €/mes

y lo que la distingue de la ley 26
  la 26 dice que el valor inicial está elegido para que la
  demostración funcione
  la 28 dice DÓNDE duele: en los servicios que facturan por
  uso, un valor por defecto no es una configuración: es una
  decisión de gasto continuo
  → y por eso pasa desapercibida: nadie revisa una casilla
    de un asistente pensando en la factura del año que
    viene                                              ley 14
```

La hipótesis de la parte 20 (clases 241 a 252, plataformas de datos e IA), escrita antes de estudiarla:

```
1. la parte 20 volverá a demostrar lo de la clase 175: en
   las cargas de datos e IA el problema dominante no será el
   modelo ni el cómputo, será el DATO —su origen, su
   permiso, su forma y quién lo escribe—; y las leyes
   dominantes serán la 21 y la 14

2. de todo lo que se monte, lo que más problemas producirá
   no será entrenar ni servir modelos, sino la ORQUESTACIÓN
   y la calidad: trabajos que fallan sin que nadie los mire
   y datos que llegan mal sin que nada lo detecte
                                              leyes 13, 15
```

```
3. los contratos de datos fracasarán por la misma razón que
   los controles: si publicar un dato con contrato cuesta
   más que publicarlo sin él, se publicará sin él  ley 16

4. en la parte de IA, la evaluación será lo que más se
   omita: se medirá la calidad del modelo en el laboratorio
   y no en producción, y la deriva se descubrirá por una
   queja, no por una señal                          ley 22

5. y una refutable con cifra: **el coste de las cargas de IA
   estará dominado por la inferencia y no por el
   entrenamiento, en una proporción de al menos tres a uno**;
   y dentro de la inferencia, por peticiones que no
   necesitaban un modelo
```

Y el cierre de la parte 19: de once clases, los tres hallazgos más graves los encontraron un inventario, una factura y otro inventario, y ninguno una alerta. La parte 20 sube una capa —a las plataformas de datos e inteligencia artificial, por encima de cualquier proveedor concreto— y empieza por la arquitectura de datos y sus contratos. Es la clase 241.

4. Comparar las tres nubes

Con el mismo sistema montado tres veces, se pueden defender conclusiones.

```
LO QUE RESULTÓ EQUIVALENTE EN LAS TRES
  el techo de disponibilidad y su aritmética   clase 185
  la teoría de colas y la concurrencia         clase 186
  la consistencia decidida por operación       clase 187
  la idempotencia y los mensajes fallidos      clase 210
  el despliegue escalonado con vuelta atrás
  la estructura del código por ciclo de vida
  el orden de implantar un control: medir, avisar, aplicar
  y las pruebas negativas, una a una

LO QUE NO
  el modelo de gobierno y su alcance
  el modelo de permisos y sus ámbitos
  la red: regional frente a global
  la observabilidad y qué viene activado
  y lo que cada una factura por uso              ley 28
```

Y la comparación medida:

```
p99 del flujo de compra         412 ms    438 ms     395 ms
disponibilidad observada       99,86 %   99,84 %    99,88 %
coste mensual                16.700 €  17.400 €   15.900 €
coste por pedido              0,046 €   0,051 €    0,044 €
valores por defecto cambiados       24        19         21
pruebas negativas fallidas       7/19      8/19       6/20
tiempo hasta producción      9 semanas 7 semanas  6 semanas
líneas de infraestructura       ~4.100    ~3.800     ~3.400
conmutación de región         12 min 40  10 min 40   8 min 15
equipo dedicado                   2,1       2,4        1,9
```

Y las cuatro conclusiones que se pueden defender:

```
1  EL COSTE DE UN SISTEMA EQUIVALENTE ES PARECIDO
   la diferencia entre la más cara y la más barata es del
   9 %, dentro de lo que cambia una decisión de diseño
   → elegir nube por precio de lista es un error
                                       clases 216, 228

2  EL TIEMPO HASTA PRODUCCIÓN BAJA CON CADA REPETICIÓN
   9, 7 y 6 semanas
   → y no por la nube: por el método
   → lo que se traslada bien es el ORDEN, no la
     configuración

3  LO QUE MÁS DIFIERE ES LO QUE MÁS CUESTA APRENDER
   el modelo operativo: gobierno, permisos, red y
   observabilidad
   → tres meses en cada nube nueva
   → y ese es el coste real de la multinube    clase 157

4  EL ERROR DOMINANTE AL AÑADIR UNA NUBE NO ES LA
```

```
   IGNORANCIA: ES EL TRASLADO
   → cinco casos documentados en esta parte
   → y la contramedida es un procedimiento, no un curso:
     al empezar con un servicio, comprobar explícitamente
     qué se hereda, qué viene activado y qué se propaga
                                              clase 229
```

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
- los perímetros están simulados un ciclo completo
- las 20 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 %
- hay comparación con las otras nubes, con cifras
- está escrito lo que no se hace

Y el cierre que enlaza con la clase siguiente: la parte 20 deja de hablar de proveedores y sube a las plataformas de datos e inteligencia artificial, que se montan sobre cualquiera de las tres. Empieza por la arquitectura de datos y sus contratos, en la clase 241.

Ejemplo trabajado

El sistema de CloudShop en producción en Google Cloud. Lo que sigue es la lista de valores por defecto cambiados, el resultado de las veinte pruebas negativas —de las que fallaron seis— y la comparación final de las tres nubes.

La lista de valores por defecto cambiados:

servicio	por defecto	cambiado a
cortafuegos	salida permitida	denegada, con permitidos
cortafuegos	selector por etiqueta	por cuenta de servicio
red	subredes automáticas	personalizadas
proyecto	cuenta de servicio por defecto	una por carga
almacenamiento	sin registro de acceso	activado donde hay datos
almacén analítico	sin límite de bytes	200 GB por consulta
almacén analítico	tablas sin partición	particionadas y agrupadas
almacén analítico	sin exigir filtro	exigido
documentos	indexa todos los campos	4 de 34
documentos	reglas permisivas de ejemplo	reglas probadas con simulador
servicio de contenedores	concurrencia por defecto usada como 1	40, medida
servicio de contenedores	sin conector de red	con conector y salida por la red
servicio de contenedores	tiempo de espera alto	3× el p99
mensajería	sin tema de fallidos	con tema y suscripción
mensajería	entrega exactamente una vez desactivada	activa donde el duplicado cuesta
registros	sin caducidad	30 y 400 días
registros	acceso a datos desactivado	activado por servicio, acotado
clúster	110 pods por nodo	32, medidos
estado del código	local	remoto, cifrado, versionado
Spanner	clave secuencial	aleatoria
compromisos	comparados con lista	con el gasto real
total		21
que funcionaban sin cambiarlos		21

Las veinte pruebas negativas: seis fallaron.

```
✓  crear recurso sin etiquetas                    rechazado
✓  crear clave de cuenta de servicio              rechazado
✓  suplantar sin permiso                          denegado
✗  acceder fuera de una condición de recurso
   → 2 de 11 servicios ignoraban la condición
                                                clase 230
✓  activar papel privilegiado sin aprobación      denegado
✓  acceso de emergencia                           90 s
✗  resolver nombre con conexión privada desde cada proyecto
   → 3 proyectos creados el mes anterior sin zonas
     asociadas: la automatización llegó después   ley 27
✓  sacar datos a otra organización            perímetro lo
                                              impide
✓  desplegar incumpliendo política            falla en
                                              validación
✓  borrar el estado y restaurarlo             4 min
✗  plan sin cambios, vacío
   → 6 recursos con deriva permanente por campos que la
     plataforma ajusta; marcados para ignorar
✓  escribir con clave secuencial              reparto
                                              correcto
✓  consultar sin filtro de partición          rechazado
✗  consultar columna sensible sin etiqueta
   → 1 vista materializada creada antes del etiquetado
     exponía la columna sin control             ley 27
✓  mismo mensaje 50 veces                     1 efecto
✓  mensaje en tema de fallidos → alerta       38 s
✓  identidad desde otro espacio de nombres    denegado
✗  perder una región
   → 8 min 15 s, dentro de lo previsto, pero la clave de
     cifrado propia vivía en una sola región y 2 almacenes
     quedaron inaccesibles                     clase 239
✗  provocar la condición de cada alerta
   → 7 de 45 no llegaron; 4 por umbral, 2 por destino y 1
     por objetivo medido en el servidor         clase 238
✓  desplegar el entorno de cero               61 min
```

Y el análisis de las seis:

```
tres por la ley 27: recursos creados antes de que existiera
  la comprobación o el etiquetado que los cubriría
dos por controles que no cubrían lo que se creía
  (condiciones ignoradas por 2 servicios, alerta con el
   indicador en el sitio equivocado)
una por una dependencia no contada en el cálculo
  (la región de la clave de cifrado)

→ y el diagnóstico es el mismo de las clases 216 y 228:
  el sistema creció y las comprobaciones no crecieron con él
→ con el añadido de que aquí ya existían las comprobaciones
  y lo que faltó fue ejecutarlas sobre lo creado después
```

Las cifras del sistema, tras tres meses:

p99 del flujo de compra	< 500 ms	395 ms
disponibilidad observada	99,8 %	99,88 %
coste mensual	15.000 €	15.900 €
coste por pedido	0,050 €	0,044 €
conmutación de región	15 min	8 min 15
coste atribuido	90 %	95 %
alertas por turno	< 2	0,6
técnicas simuladas detectadas	> 90 %	13/14
despliegue del entorno de cero	—	61 min

La comparación final de las tres nubes, con lo que costó de verdad:

coste mensual	16.700 €	17.400 €	15.900 €
coste por pedido	0,046 €	0,051 €	0,044 €
p99 del flujo	412 ms	438 ms	395 ms
conmutación de región	12 min 40	10 min 40	8 min 15
valores por defecto cambiados	24	19	21
pruebas fallidas	7/19	8/19	6/20
tiempo hasta producción	9 semanas	7 semanas	6 semanas

```
meses para aprender el modelo
  operativo                          —         3          3
equipo dedicado                  2,1       2,4        1,9

Y OPERAR LAS TRES A LA VEZ
  equipo dedicado sumado                          6,4
  equipo real necesario                           7,8
  → un 22 % más que la suma, por los cambios de contexto
    y por mantener tres modelos operativos al día
                                                clase 157
```

Y las decisiones que se tomaron con esta comparación:

```
no migrar nada entre nubes: los tres sistemas funcionan
no operar los tres a la vez para la misma carga
  → CloudShop mantiene AWS como principal, Azure por el
    contrato corporativo y Google para las cargas de datos
    y de IA                                     clase 157
y el criterio escrito
  «cada nube por un motivo declarado; ninguna por
   preferencia»
```

La lección que este proyecto deja: veintiún valores por defecto cambiados, todos los cuales funcionaban, y los cinco más caros estaban en servicios que facturan por uso. De las seis pruebas negativas fallidas, tres fueron por recursos creados después de que existiera la comprobación. Y comparando las tres nubes con el mismo sistema montado bien, la diferencia de coste es del nueve por ciento: la nube no decide el coste; lo deciden las decisiones de diseño y los valores por defecto que nadie revisa.

Laboratorio guiado

Ejecuta desde la raíz:

```
python classes/part-19-gcp-production-architecture/240-proyecto-cloudshop-productivo-en-google-cloud/
lab.py
```

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

1. Lee exercise.steps y formula una predicción antes de ejecutar.
2. Ejecuta la práctica y verifica todos los elementos de checks.
3. Provoca el caso de negativetest y explica la señal observada.
4. Materializa cloudshop-gcp 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-gcp para el caso CloudShop. Incluye una alternativa descartada, un supuesto que pueda falsarse, una prueba de fallo y una decisión de rollback.

■ Criterio de aceptación

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

- [ ] Otra persona puede repetir el recorrido sin conocimiento tácito.

■■ Errores frecuentes

Síntoma	Causa probable	Corrección
Al añadir una nube nueva fallan cosas que en las anteriores funcionaban	Se trasladaron suposiciones del modelo operativo anterior	Al empezar con cada servicio, comprueba explícitamente qué se hereda, qué viene activado y qué se propaga; es un procedimiento, no un curso.
Los valores por defecto caros pasan desapercibidos	En los servicios que facturan por uso, un valor por defecto es una decisión de gasto que no parece una decisión	Revisa primero los valores por defecto de todo lo que se factura por uso, y estima su coste antes de aceptar el asistente.
Las comprobaciones pasan y aparecen huecos en lo creado después	La automatización llegó más tarde que los recursos	Toda automatización nueva se acompaña de una tarea de remediación y de una comprobación periódica sobre lo existente.
Se elige nube comparando precios de lista	El coste lo deciden el diseño y los valores por defecto, no el proveedor	Compara el coste del mismo sistema montado bien; la diferencia suele estar dentro del ruido.
Operar varias nubes consume más equipo que la suma de las partes	Se cuentan las tecnologías y no los modelos operativos ni los cambios de contexto	Justifica cada nube por un motivo declarado y cuenta el coste operativo real, que es superior a la suma.
Una clave de cifrado propia deja datos inaccesibles al perder una región	La clave vive en una región y no entró en el cálculo del techo	Replica la clave y cuenta su disponibilidad como una dependencia dura más.

■ 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 228 falló y qué patrón revela su fallo?
2. ¿Qué dice la ley 28 y en qué se distingue de la ley 26?
3. ¿Qué distinción faltaba en la predicción sobre identidad?
4. ¿Qué resultó equivalente entre las tres nubes y qué no?
5. ¿Por qué operar tres nubes cuesta más que la suma de operar cada una?

Referencias

- Google Cloud (2025). Architecture Framework. <https://cloud.google.com/architecture/framework>
- Google Cloud (2025). Enterprise foundations blueprint. <https://cloud.google.com/architecture/security-foundations>
- Google Cloud (2025). Landing zone design. <https://cloud.google.com/architecture/landing-zones>
- Beyer, B. y otros (2018). The Site Reliability Workbook. <https://sre.google/workbook/table-of-contents/>
- Forsgren, N. y otros (2018). Accelerate — medir antes y después con las mismas definiciones. <https://itrevolution.com/product/accelerate/>
- Documentación oficial vigente del servicio implementado; registra URL y fecha de consulta.

Descargar: Parte 19 en PDF · Recorrido de Google Cloud en PDF · Manual integral

Anterior	Índice	Siguiente
← 239 · SCC, VPC Service Controls, KMS y FinOps	Parte 19 · Programa	241 · Lakehouse, warehouse, mesh y contratos de datos →

Evaluación — 240 Proyecto: CloudShop productivo en Google Cloud

Preguntas

1. Explica el problema que resuelve esta capacidad sin mencionar un proveedor.
2. Identifica una frontera de responsabilidad y su propietario.
3. Compara dos alternativas mediante confiabilidad, seguridad, costo y operación.
4. ¿Qué demuestra el laboratorio y qué no puede demostrar?
5. Propón un caso límite o una prueba de fallo.

Reto verificable

Entrega cloudshop-gcp 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.