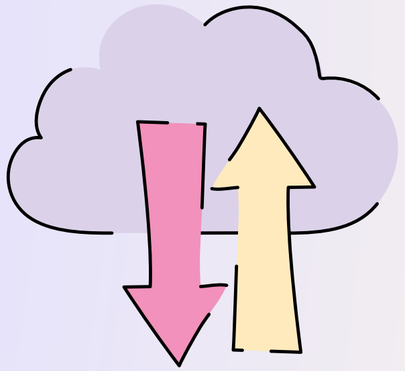
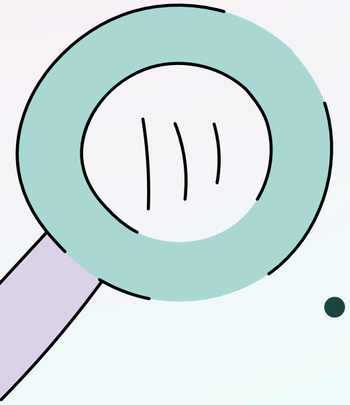
Universidad Nacional de Ingenieria



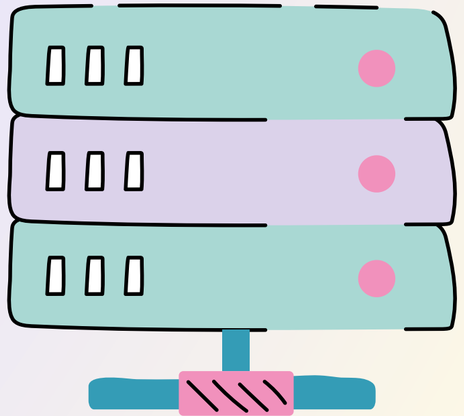
ELECCIÓN DE PATRONES PARA ESCENARIO COMPLEJO (MULTI-CLOUD)

Autor: Edy Saul Serrano Arostegui
Curso: Desarrollo de software
Actividad: 15
Profesor: Cesar Lara Avila

Problema y Contexto

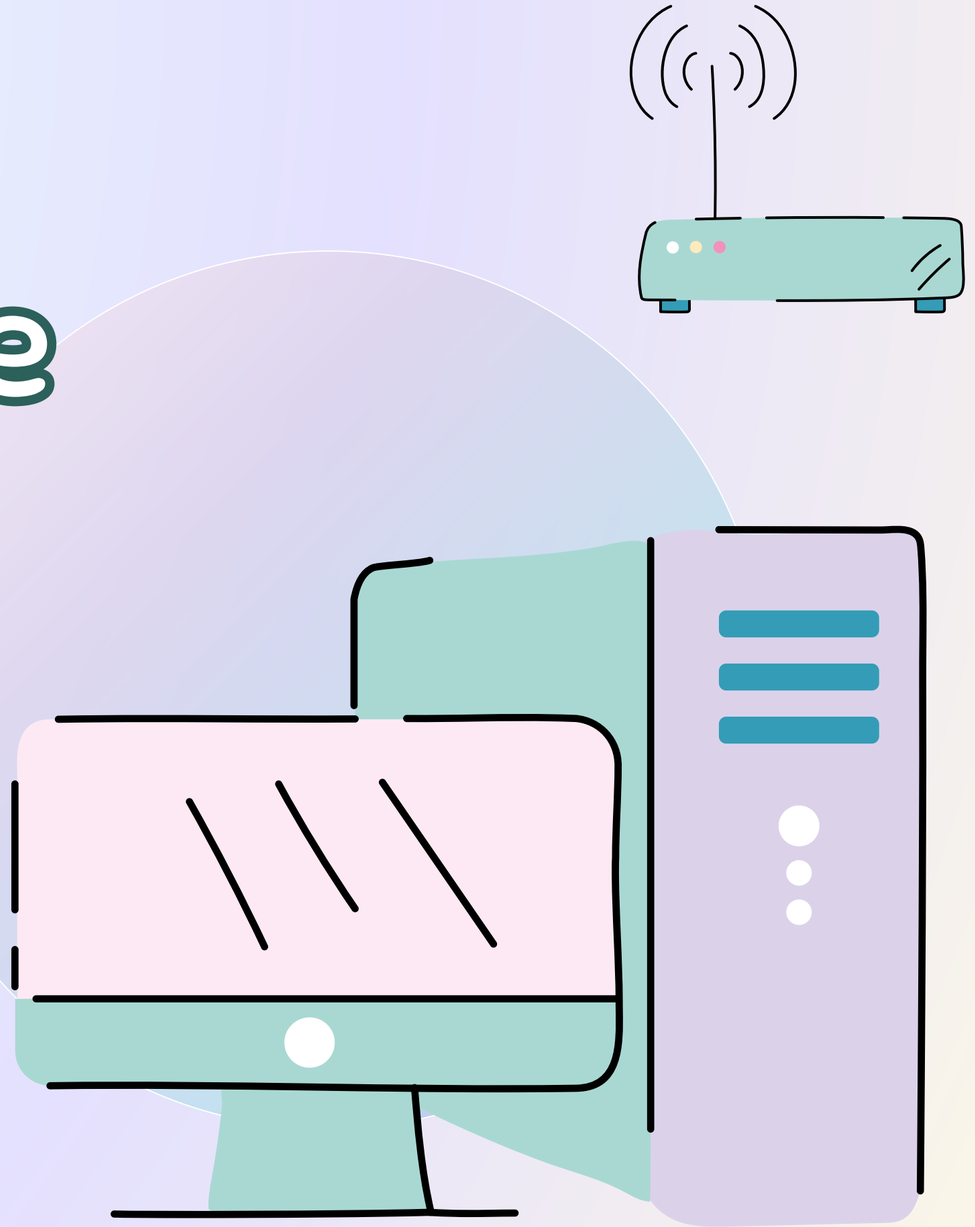


- **Escenario:** despliegue multi-cloud (GCP + AWS) con networking, servidores y seguridad.
- **Retos:** nombres distintos, APIs/proveedores diferentes, orquestación cruzada, requisitos de seguridad.
- **Objetivo:** diseñar una forma de exponer recursos reutilizables y estable para consumidores.



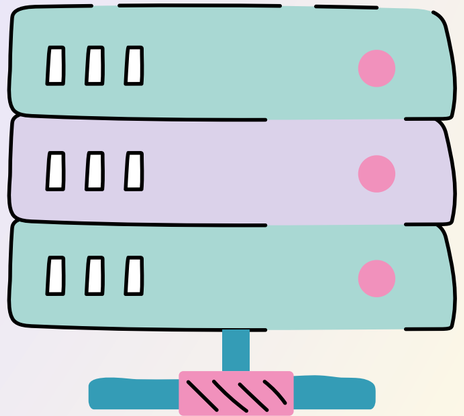
Restricciones clave

- Autonomía de equipos por cloud.
- Necesidad de contratos estables (outputs) para consumidores.
- Posible evolución independiente (versionado por módulo).
- Requisitos de gobernanza y seguridad corporativa.



Patrón(s) recomendados

- **Usar combinación:** Adapter (por proveedor) + Facade (contrato estable) + Mediator (cuando la orquestación entre clouds es compleja).
- **Motivo:** Adapter normaliza diferencias; Facade expone interfaz única; Mediator coordina flujos multi-cloud.



Razonamiento y trade-offs

Adapter

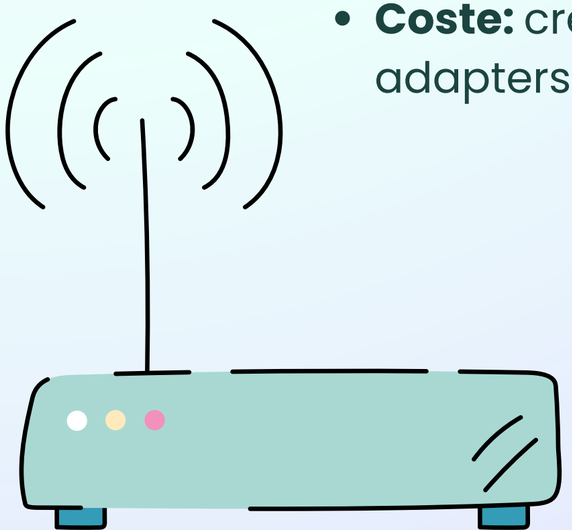
- **Ventaja:** desacopla consumidores de implementaciones cloud-specific.
- **Coste:** crear/maintain adapters por proveedor.

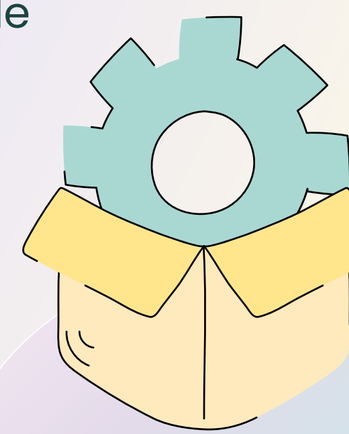
Facade

- **Ventaja:** proporciona contrato estable y reduce superficie de cambio.
- **Coste:** punto único de evolución / versión.

Mediator

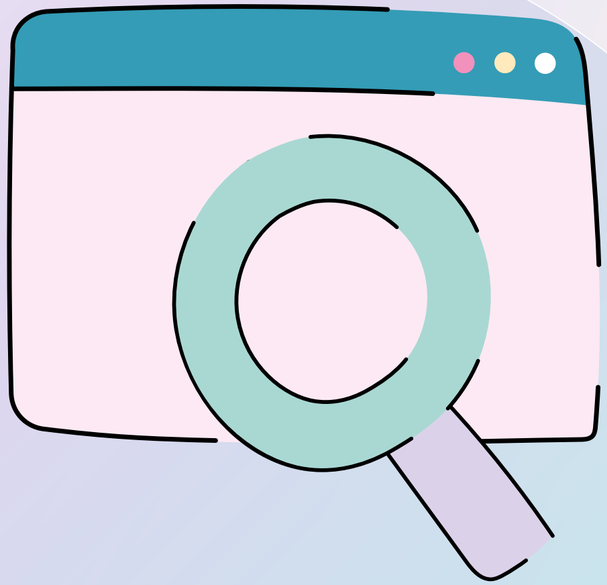
- **Ventaja:** orquesta interacciones complejas (failover, sincronización).
- **Coste:** mayor complejidad y riesgo de "god object".

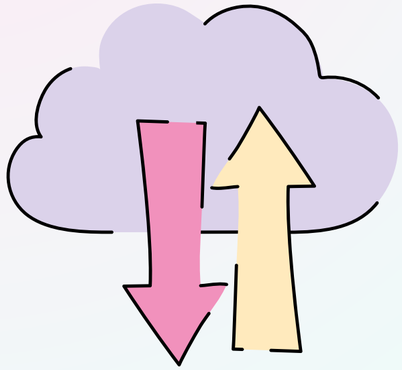




Riesgos y mitigaciones

- **Riesgo:** facade se convierte en cuello de botella → Mitigar con versionado y contratos claros.
- **Riesgo:** adapters duplican lógica → Mitigar con tests y utilidades comunes.
- **Riesgo:** Mediator con demasiada lógica → limitar responsabilidades y escribir tests unitarios.





GRACIAS!!!