

PATRONES DE ARQUITECTURA DE SOFTWARE

CQRS Y EVENT SOURCING

Daniel Blanco Calviño

COMANDOS Y CONSULTAS

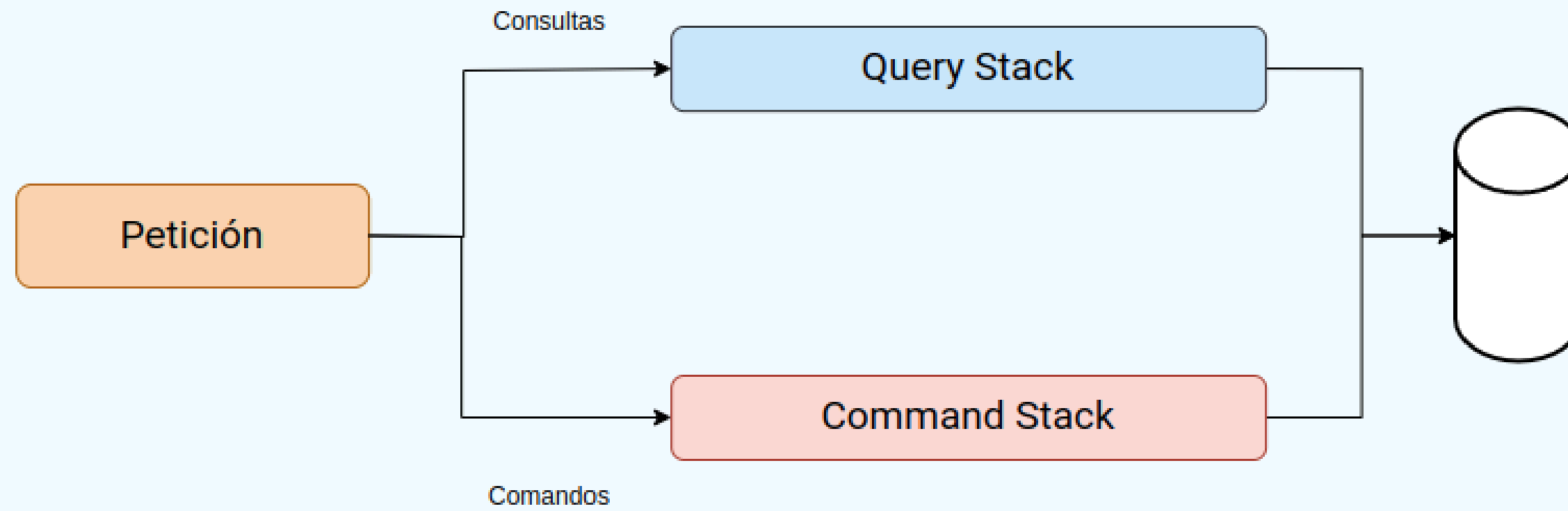
- **Comandos.**

- Acciones que **realizan una modificación en el estado del sistema** y que no devuelven información.
- Ejemplo: Escribir un comentario en un post.

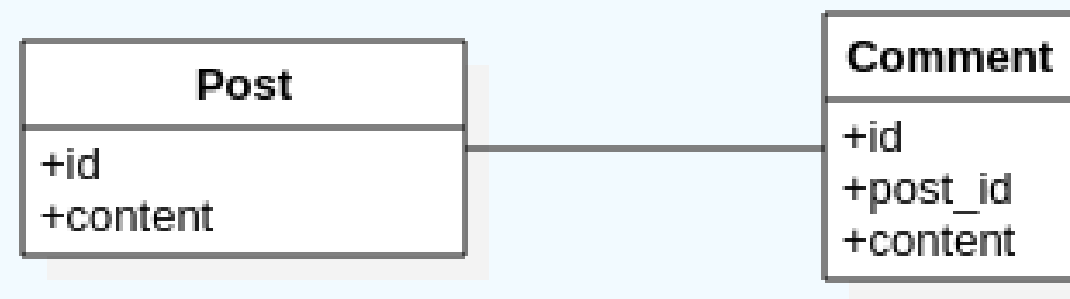
- **Consultas.**

- Acciones que **no alteran el estado del sistema**, tan sólo devuelven datos.
- Ejemplo: Obtener los últimos diez comentarios de un post.

CQRS



EJEMPLO DE UN BLOG



REPRESENTACIÓN RELACIONAL

POST			COMMENT		
id	content		id	post_id	content
1	Post Content 1		1	1	First Comment
2	Post Content 2		2	1	Second Comment
3	Post Content 3		3	2	Third Comment



- **Rápido insertando** nuevos comentarios.



- **Lento leyendo** el número de comentarios de cada post.
- **Lento leyendo** el contenido de los comentarios de un post.

REPRESENTACIÓN NOSQL

```
1 {
2   "posts": [
3     {
4       "id": 1,
5       "content": "Post Content 1",
6       "numberOfComments": 2,
7       "comments": [{
8         "id": 1,
9         "content": "First Comment"
10      },
11      {
12        "id": 2,
13        "content": "Second Comment"
14      }
15    ]
16  },
17  {
18    "id": 1,
19    "content": "Post Content 1",
20    "numberOfComments": 2,
21    "comments": [{
22      "id": 3,
23      "content": "Third Comment"
24    }]
25  },
26  {
27    "id": 3,
28    "content": "Post Content 2",
29    "numberOfComments": 0,
30    "comments": []
31  }
32 ]
33 }
```

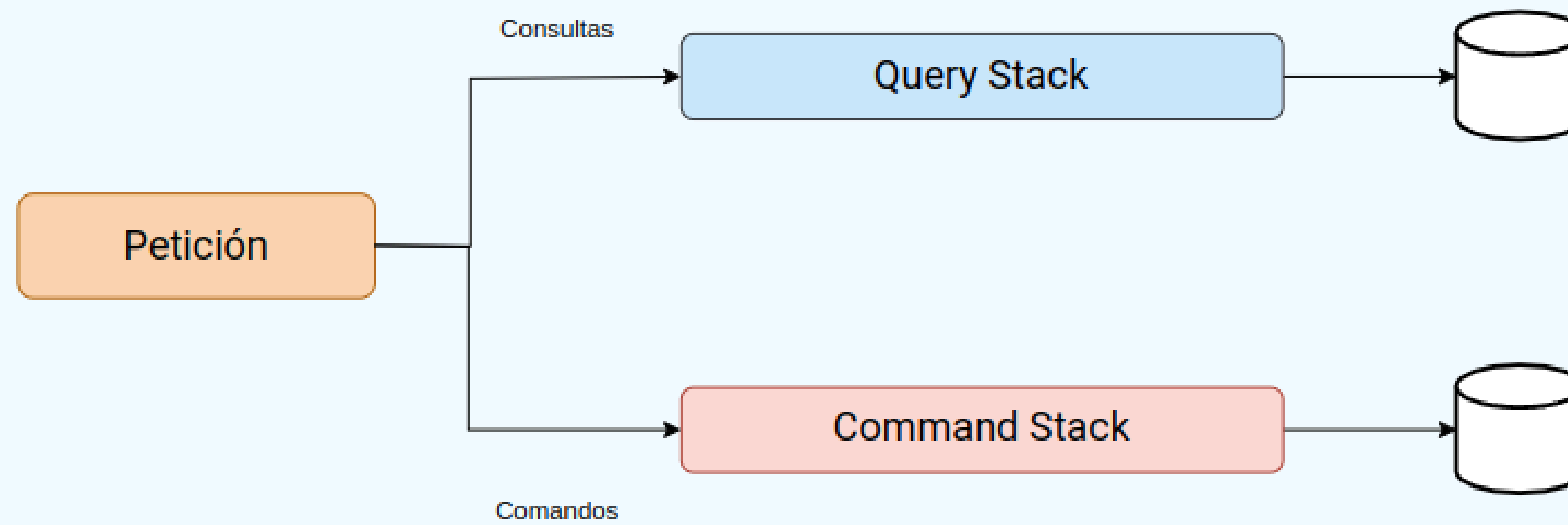


- **Rápido leyendo** los posts, el número de comentarios y su contenido. Información desnormalizada.

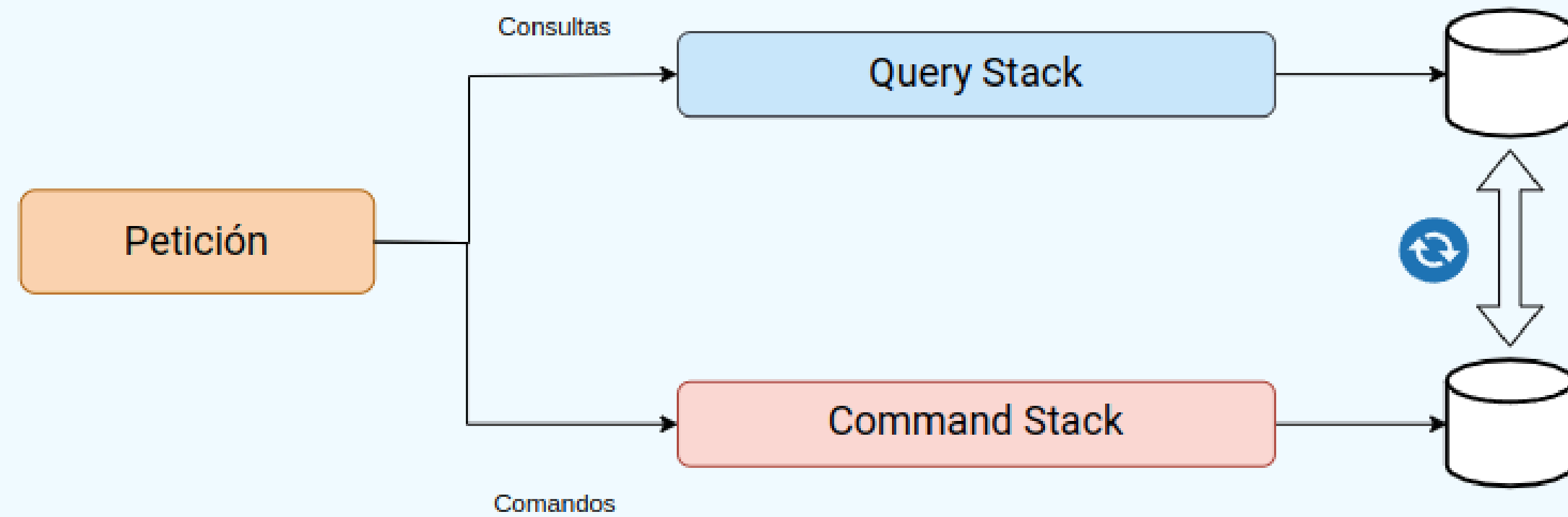


- **Muy lento** añadiendo comentarios nuevos

CQRS CON BBDD INDEPENDIENTES



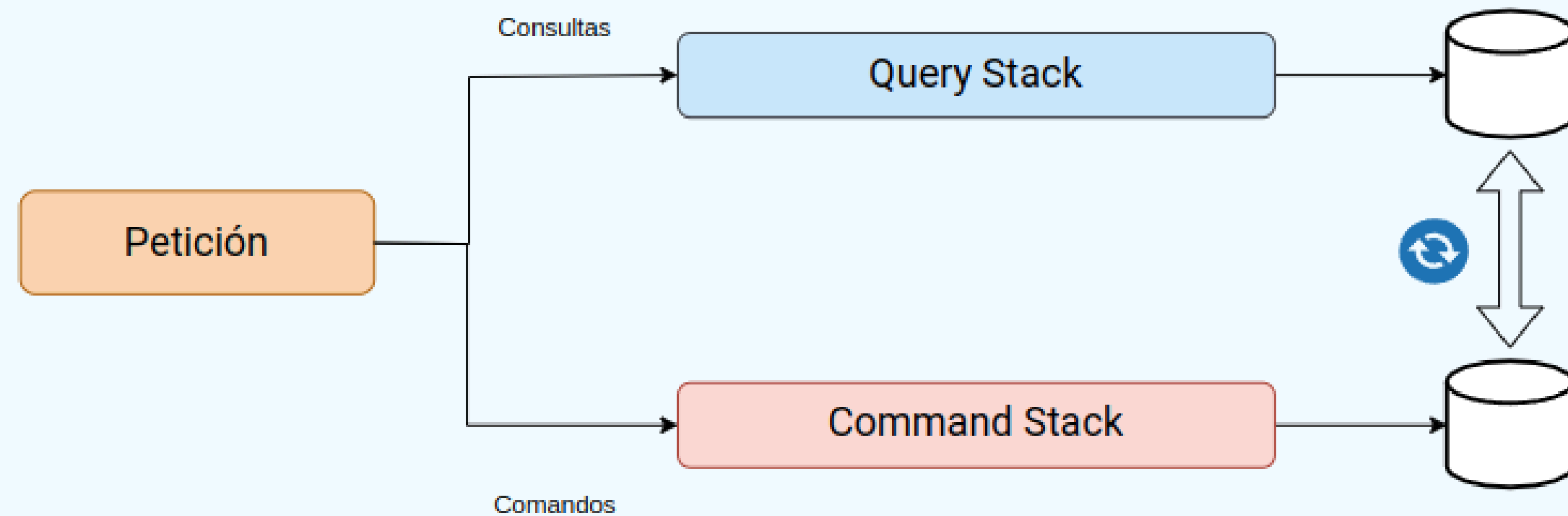
EL GRAN PROBLEMA, LA SINCRONIZACIÓN



MÉTODOS DE SINCRONIZACIÓN

- **Consistencia inmediata.**
 - Método síncrono.
- **Consistencia eventual.**
 - Sincronización asíncrona.
- **Consistencia programada.**
 - Sincronización a ciertas horas del día.
- **Consistencia bajo demanda.**
 - Se realiza la sincronización cuando se determine necesaria.

EL GRAN PROBLEMA, LA SINCRONIZACIÓN



- **Dependencia directa** con el mecanismo de sincronización.
- Complejidad para insertar el modelo de escritura en la BBDD de lectura.

EVENT SOURCING

- Sistemas que almacenan el **estado** como una **secuencia de eventos**.

EVENT SOURCING

- Sistemas que almacenan el **estado** como una **secuencia de eventos**.

1. Añadir post
2. Añadir comentario
3. Modificar post
4. Eliminar comentario

Modelo tradicional

POST SIN COMENTARIOS

Event Sourcing

AÑADIR POST #1

AÑADIR COMENTARIO #1

MODIFICAR POST #1

ELIMINAR COMENTARIO #1

EVENT SOURCING + CQRS

- **Independencia del sistema de sincronización.**
 - No es necesario realizar una llamada directa para activarlo.
 - Command Stack genera eventos y el sistema de sincronización los consume.
- **Las traducciones de eventos son más sencillas.**
 - Traducir una entidad completa es más complejo que traducir un evento.
 - Podemos elegir desde dónde sincronizar.

ASPECTOS POSITIVOS



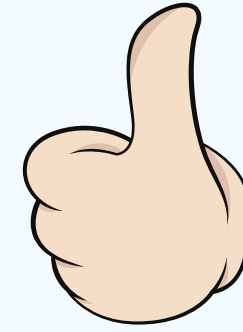
- **Tratamiento independiente de las lecturas y las escrituras.**
 - Posibilidad de tener dos modelos distintos.
 - Uso de bases de datos adecuadas para cada situación.
 - Escalado independiente en función de las necesidades de lectura / escritura.
 - Normalización y desnormalización independientes.
- **Posibilidad de tener dos equipos independientes.**
 - Un equipo para el stack de comandos.
 - Otro equipo para el stack de consultas.

ASPECTOS NEGATIVOS



- **Gran complejidad**
 - Mantenimiento de dos stacks distintos para escritura y lectura.
 - Mantenimiento de múltiples Bases de Datos.
- **Sincronización**
 - Mantener la consistencia de los datos es un problema añadido.
- **Duplicidad / redundancia de código**

CUÁNDO USAR



- Sistemas dónde la **escalabilidad es muy importante o crítica**.
- **No es necesaria una consistencia inmediata** de la información.
 - En caso contrario, no sólo no ganamos nada, sino que será perjudicial.
- Distinto tratamiento para las escrituras y lecturas.