



## Qué no es un Command (en CQRS)

Estamos acostumbrados que al escuchar **Command** nos venga a la cabeza un **comando de CLI**, pero en el contexto de CQRS no es así.

También se nos puede pasar por la cabeza la definición de **Command del GoF** ([libro Design Patterns](#)), una forma de encapsular la lógica de nuestro dominio en una clase autocontenida que sea autoejecutable, pero en este caso, tampoco es así.

## Qué es un Command

En CQRS, un Command representa la intención de realizar una operación en nuestro sistema que acabe modificando el estado de tal.

## Cuando nos viene bien utilizarlo

Muchas veces se dice que no has de hacer CQRS si lo que vas a montar es un blog, pero como todo en esta vida, depende.

Si vas a construir o gestionas un blog con muchísimo tráfico, CQRS encaja muy bien dentro de tu aplicación, ya que por términos de rendimiento nos aporta de gratis poder realizar tareas asíncronas (usando una implementación asíncrona del CommandBus).

También nos viene muy bien trabajar con este patrón cuando tenemos equipos grandes, ya que al tener separado lo que son intenciones de consultas nos ahorraremos muchos [merge hell's](#).)

## Ejemplo

Fuera de la teoría, el Command en sí no es más que un DTO (objeto de transferencia de datos) el cuál representa la acción que queremos hacer. Por ejemplo, si quisiéramos cambiar el nombre de un usuario podríamos tener un Command tal que:

```
final class RenameUserCommand extends Command
{
    private $userId;
    private $newName;
    public function __construct(string $userId, string $newName)
    {
        $this->userId = $userId;
        $this->newName = $newName;
    }
    public function userId(): string
    {
        return $this->userId;
    }
    public function newName(): string
    {
        return $this->newName;
    }
}
```

Este Command nos llegaría a nuestro CommandHandler, el cuál crearía dos **Value Objects: UserId y UserName** (de esta forma estaríamos validando que los valores del Command cumplen con los requisitos de nuestro dominio).

Luego el CommandHandler enviaría estos datos al caso de uso, el cuál ya se encargaría de ir al Repositorio para recuperar el usuario, cambiarle el nombre, volverlo a persistir y luego lanzar el evento de nombre modificado.

## Enlaces relacionados

- [Is a CQRS Command = to a GoF Command?](#)
- [Ejemplo de Command](#)

## Conclusión

Utilizar Commands nos ayuda a reutilizar el código de una manera muy sencilla. También hemos visto que se adapta muy bien cuando tenemos aplicaciones que exigen un alto rendimiento y cuando tenemos equipos grandes.

Next >

🏆 You earned 100/100pts · 🔄 Reset

#### Group Discussion

Start a discussion

- **SD** Muchos datos en el command  
Santiago D. 3 days ago 🗨️ +0 -0 [Answer >](#)
- **GD** error en parametro del command  
Giuseppe D. 7 months ago 🗨️ +1 -1 [Answer >](#)
- **LW** Resource not found  
Leandro W. a year ago 🗨️ +2 -1 [Answer >](#)

