

КОГДА МОЖНО ПОЖЕРТВОВАТЬ ПРИНЦИПОМ DRY

Чётко понимайте что вы делаете, если необходимо пожертвовать DRY, а также хорошо обосновывайте такое решение. Вот несколько примеров, когда отход от принципа оправдан:

- **при использовании клиент-серверных приложений.**
В микросервисной архитектуре практически всегда используются повторяющиеся классы в разных сервисах. Различаться они будут своими контекстами, поскольку их доменные ответственности будут отличаться. В этом случае, не нужно делать мега сервис, который занимается конкретной сущностью, такое дублирование является естественным

Точно такие же проблемы происходят в коде, который находится на backend и frontend. И там и там будут жить одни и те же сущности и это нормально