

no caso de tolerância a falhas, esse modelo também torna-se passível de recuperação e evidência caso o operador com perfil de desenvolvedor, por exemplo, altere dados que não deveria, pois será possível identificar essas alterações mesmo para esse perfil.

isso garante segurança em testar conceitos no ambiente de produção. Pensando aqui, poucos cenários seriam testados em outros ambientes, fazendo com que a necessidade de haver esses outros ambientes venha a ser discutida.

## O próprio sistema permite auditoria nesse modelo

recuperação de falhas e motivos para acreditar que o ambiente de produção possa existir sozinho nesse modelo:

1. estou garantindo reprocessamento de dados nos processos por armazenar cada estado de processo e permitir esse replay começando por qualquer processo dentro do plano de execução
2. a imutabilidade da persistência me permite segurança quanto a consistência dos dados
3. cada estado persistido também tem quem e o que foi responsável por essa ação
4. componentização de ponta a ponta me permitindo ter versões diferentes de processos e planos de execução por perfil de usuário na maneira que for necessário

Se cada processo que altera e consolida estado na persistência é realmente imutável e único dentro de um plano de execução, então numa recuperação de dados independentemente de ser composição ou agregação, essa recuperação pode ser feita obtendo o estado que cada processo persistiu de forma independente e unida ao fim para compor o resultado.

## Ficar bem ciente das diferenças:

- agregação os dados fazem sentido em conjunto ou não independente do contexto
- composição os dados fazem sentido apenas quando em conjunto de acordo com contexto apresentado

Essa modelagem se trata de appendonly, logo, nenhum dado jamais será alterado ou removido na persistência. O que pode acontecer é mover os dados para outro servidor e banco de dados quando a informação não for mais útil.