# Desafio VR Desenvolvimento

1. Recebemos um código desenvolvido por terceiros de um sistema que possui alto volume de lógica de negócio e apresenta as seguintes características:

1 - O sistema recebe requisições REST, está dividido em camadas e possui classes de domínio;

2 - O controller recebe a requisição e está com toda lógica de negócio. Monta e repassa o domínio para a aplicação;

3 - A aplicação tem a responsabilidade de repassar o objeto pronto para o repositório;

4 - O repositório apenas persiste os objetos mapeados do hibernate através de spring data;

5 - O domínio apenas faz o mapeamento para o BD;

6 - Nenhum teste unitário foi escrito.

7 - O sistema está escrito em java para rodar como spring boot.

Apresente observações/problemas sobre essa solução.

Comente qual(is) a(s) sua(s) estratégia(s) para melhorar este sistema em termos de qualidade e manutenção. Justifique suas decisões.  
  
Resposta:  
1 – a princípio, normal.  
2 – se toda a lógica está nos controllers, está bagunçado o código, recomendável dividir este código em controller e serviço.  
3 – ok

4 - dependendo da forma eu foi feito, não vejo problema

5 – Não vejo problema.

6 – Problema!

7 - ok

1. Descreva quais são as principais limitações ao se adotar servidores de aplicação em uma arquitetura orientada a microsserviços.

Resposta:  
Um servidor de aplicação carrega por si so um peso em configuração e controle de transação. Recomenda-se leveza e ninguém no caminho entre a lógica da aplicação e sua aplicação final.

1. Atualmente, diversas aplicações escritas em Java estão deixando de serem desenvolvidas para rodarem em servidores (JBoss, Tomcat), adotando ferramentas que disponibilizam um servidor embutido na própria ferramenta. Quais são os principais desafios ao se tomar uma decisão dessas? Justifique sua resposta.

Resposta:   
Refatorar código existente par não precisar de recursos do server, taais como controle do cilco de vida dos objetos, acesso a recursos e transações.

1. Teste prático (em anexo)