# 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:

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

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

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

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

- O domínio apenas faz o mapeamento para o BD;

- Nenhum teste unitário foi escrito.

- 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.  
**R**:Em primeiro lugar tiraria a classe de domínio da controller, criaria um dto para parte que tem interação com usuário final, mudaria toda logica para classe de aplicação mudaria nome para uma classe de service, também na classe service teria uma função para converter classe de domínio para dto e assim passar esse dto para classe de controller, criaria todos os testes unitário de todas as camadas. Acredito que com essas mudanças o código estaria no padrão para aplicações REST.

1. Descreva quais são as principais limitações ao se adotar servidores de aplicação em uma arquitetura orientada a microsserviços.  
   **R:**
2. 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.  
   **R**: Acredito que o principal desafio é na manutenção e monitoramento por que com servidor embutido, as aplicações ficam isolada e em uma arquitetura com vários micros serviços, é preciso configurar cada microserviço individualmente
3. Teste prático (em anexo)