# 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:

- Trabalhar com Rest foi uma boa escolha, mas colocar toda a camada de negócios no Controller é bem questionável, Para melhorar esse código podemos começar por aqui, tirando todo o tratamento de negócio e colocando em classes Services.

- Identificar tratamentos especializados, por exemplo tratamento de datas ou cpfs, colocando-os em classes utilitárias.

- O controller ficará responsável somente por receber os dados e em chamar Services passando o dado bruto recebido.

- Criar DTOs para tratamento dos dados recebidos também será necessário.

- Os testes unitários são necessários para garantir a qualidade do código no presente e futuramente assegurar que ele não será quebrado, por isso é necessário refatorar o código para incluí-los. Nas próximas implementações será uma boa prática desenvolver com TDD.

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

R: O primeiro problema que me vem a mente em adotar essa estratégia é a disponibilidade pois todas as aplicações estarão em um único servidor. Mesmo pulverizando isso em mais máquinas ainda há o problema de espaço físico e financeiro. Para micro-serviços, o ideal seria optar por uma arquitetura em nuvem, de forma a aumentar a performance e escalabilidade das aplicações.

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.

R: Um dos desafios dessa implementação é que as configurações de servidor de aplicação passam a ser individuais. Neste modelo é preciso que cada aplicação configure seu tratamento de logs, monitoramento e segurança. O tratamento de atualizações é importante também nesse modo de utilização.

Não pode-se compartilhar, entre as aplicações, os recursos utilizados. Precisamo torna-lá stateless.

Talvez a principal desvantagem deste modelo seja o fato de ser necessário restartar a aplicação, para qualquer alteração feita no projeto.

1. Teste prático (em anexo)