# 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.

Resposta.

O interessante seria usar as interfaces para usar a camada de service com as regras de negócio, para diminuir a dependência na classe controle e estabelecer um contrato do que deve ser implementado. O spring boot possibilita através do uso de injeção de dependência o uso dessas interfaces, onde posso sempre alterar a implementação, mas sempre se saberá o que deve ser implementado. Possibilita também aplicar os princípios solid com classes com responsabilidades únicas e métodos. Facilita o uso dos testes unitários. Interessante também o uso dos dto’s para representar o body passado na api rest e não alterar as classes de domínio. O controle deve apenas repassar e receber as informações deixando as classes service tratarem os dados e as regras de negócio.

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

Resposta

Servidores de aplicação são muito complexos envolvendo vários mecanismos e a arquitetura orientada a microsserviços requer mais um isolamento de componentes em processos próprios, onde eles podem ser executados em containers visando uma maior mobilidade e possuindo particularidades inerentes ao seu negócio.

O uso de um servidor de aplicação para cada app de microsserviço adiciona uma sobrecarga. Um número maior de instâncias serão necessários comparado ao modelo de configuração tradicional. Microsserviços aumentam o número de componentes que devem ser aplicados e aumentam também a complexidade para manter o servidor de aplicação.

Outro problema no servidor de aplicação e que se uma aplicação der erro no servidor vai afetar as outras 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.

Resposta.

A busca por mecanismos de monitoramento, deploy e controle de transações, uma menor limitação de ajustes e tuning no servidor. Uma quantidade maior de serviços requer mais organização, pois os mesmos estão bem espalhados. Aplicações desenvolvidas para rodarem num servidor externo a aplicação possibilita uma maior concentração das funcionalidades ao contrário de usar servidores embutidos.

Com o servidor de aplicação, há mais uma etapa nos testes com o deploy do arquivo war, com embedded server não existe a etapa de compressão e descompressão. A configuração de um servidor de aplicação e bastante complexa, dificultando as ferramentas de automação e essa configuração deve ser feita em todos os ambientes como dev, teste, qa e prod.

1. Teste prático (em anexo)