# Desafio VR Desenvolvimento

1. Recebemos um código desenvolvido por terceiros de um sistema que possui volume alto de lógica de negócio e apresenta a seguinte característica:

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 a sua estratégia para melhorar este sistema em termos de qualidade e manutenção.

1. Descreva quais são as principais limitações ao se adotar servidores de aplicação em uma arquitetura orientada a microsserviços.
2. Quais são os principais desafios ao se adotar aplicações do tipo "Embedded Servlet” em relação a aplicações desenvolvidas para um application server?
3. Defina os resources e métodos http para o controller abaixo utilizando annotation do Spring MVC:

@RestController

public class PedidoResource {

public ResponseEntity<ServiceResponse<PedidoVO>> consultar(String idPedido) {

...

}

public ResponseEntity<ServicePageableResponse<List<PedidoVO>>> listar(PedidoListarDto pedidoListarDto) {

...

}

public ResponseEntity<ServiceResponse<Void>> criar(PedidoVO pedido) {

...

}

public ResponseEntity<ServiceResponse<Void>> alterar(

String idPedido,

PedidoVO pedido) {

...

}

public ResponseEntity<ServiceResponse<Void>> excluir(

String idPedido,

PedidoVO pedido) {

...

}

public ResponseEntity<ServiceResponse<Void>> atualizarStatus(

String idPedido,

Status status) {

...

}

public ResponseEntity<ServiceResponse<Void>> atualizarVencimentoPedido(

String idPedido,

VencimentoPedidoDto vencimentoPedidoDto) {

...

}

}