

VLADMIR BARBEIRO DA CRUZ

POSTECH

ARQUITETURA E
DESENVOLVIMENTO JAVA

WELCOME TO ARQUITETURA E DESENVOLVIMENTO
JAVA

FASE 3

Tech Challenge

Tech Challenge é o projeto que englobará os conhecimentos obtidos em todas as disciplinas da fase. Esta é uma atividade que, em princípio, deve ser desenvolvida em grupo. Importante atentar-se ao prazo de entrega, pois trata-se de uma atividade obrigatória que vale nota em todas as disciplinas da fase.

O problema

Criar um sistema de Reserva e Avaliação de Restaurantes.

Funcionalidades Básicas:

1. **Cadastro de Restaurantes:** Os restaurantes podem se cadastrar no sistema, fornecendo informações como nome, localização, tipo de cozinha, horários de funcionamento e capacidade.
2. **Reserva de Mesas:** Os usuários podem fazer reservas para datas e horários específicos.
3. **Avaliações e Comentários:** Após a visita, os usuários podem avaliar o restaurante e deixar comentários sobre sua experiência.
4. **Busca de Restaurantes:** Os usuários podem buscar restaurantes por nome, localização ou tipo de cozinha.
5. **Gerenciamento de Reservas:** Os restaurantes podem gerenciar as reservas, visualizando e atualizando o status das mesas.

O que será avaliado

Clean Architecture:

1. **Design Arquitetônico:** Estruturar o backend de acordo com a Clean Architecture, definindo entidades, regras de negócio, adaptadores de interface e frameworks.
2. **Clean Code em Testes:** Implementar práticas de código limpo em todas as fases do desenvolvimento, com foco especial em testes.

Qualidade de Software:

1. **Testes Unitários (TDD com JUnit por exemplo):** Desenvolver o backend utilizando TDD, focando em cobrir todos os aspectos críticos da aplicação.
2. **Testes Integrados e Inspeção de Código:** Realizar testes integrados para garantir que todas as partes do sistema funcionem juntas, e usar ferramentas de inspeção de código para manter a qualidade.
3. **Testes de Integração no Controller, CI e BDD:** Testar os controllers para garantir que a lógica de negócios esteja correta. Integrar os testes no pipeline de CI e adotar BDD para melhorar a comunicação entre a equipe.
4. **Testes Não Funcionais:** Incluir testes de carga e desempenho para garantir que o sistema possa lidar com um grande número de usuários e reservas.
5. **Medir cobertura de teste:** Usar o coverage para metrificar a porcentagem de testes realizados.

Deploy:

Ambientes Diversos de deploy: Realizar deploy do sistema em diferentes ambientes, incluindo local, AWS ECS, Azure e uma plataforma de nuvem gratuita como Heroku, Render, entre outros.

Boa sorte!



POSTECH