Estágio de Desenvolvimento - Desafio Técnico

Prepare-se para colocar suas habilidades em Java e SQL à prova neste desafio. Seu objetivo será criar um pequeno sistema de gerenciamento de reservas para o fictício hotel “Eclipse Hotel”. Você terá a oportunidade de demonstrar sua capacidade de desenvolvimento utilizando o poderoso framework Spring Boot, vamos lá?

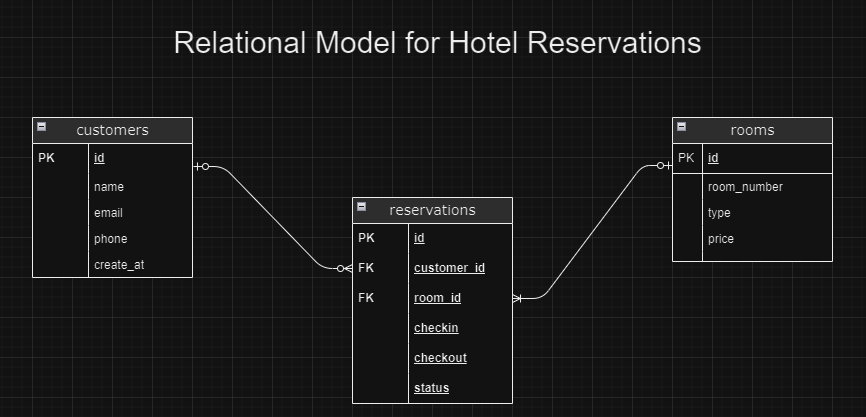
Antes de começarmos, seguem algumas instruções:

* Este desafio consiste em 2 etapas, a primeira de desenvolvimento Java e a segunda de banco de dados.
* Você deverá possuir uma conta no Bitbucket [Bitbucket | Git solution for teams using Jira](https://bitbucket.org/product/). Caso não possua, a criação é gratuita.
* Você deverá criar um novo repositório no bitbucket no seguinte formato: (tecnologia)(nível da vaga)(nome)(sobrenome).   
  ex: javaestagiariofulanodetal  
  E realizar todos os commits dentro do prazo estipulado.
* Após a criação do repositório, deverá liberar acesso a nossa conta de seleção: [selecao@mobicare.com.br](mailto:selecao@mobicare.com.br) e compartilhar o link do repositório conosco.
* **Para o desafio de SQL, deverá ser commitado 1 arquivo no formato .sql na raíz do projeto, contendo todas as instruções solicitadas ou as que você conseguir solucionar dentro do prazo.**
* **Caso não consiga finalizar o projeto por inteiro, realize o commit das etapas que concluiu.**
* **Este documento possui 3 páginas.**

# Etapa 1 - Avaliação de desenvolvimento

Requisitos utilizados para o desenvolvimento:

* Java 8 ou superior
* Spring Boot
* Jpa/Hibernate
* PostgreSql ou H2 em memória



Acima apresentamos o modelo lógico do banco de dados disponibilizado por nosso DBA. Utilizaremos o banco de dados PostgreSql ou outro banco relacional de sua preferência, para este desenvolvimento.

Atenção:

Na tabela reservations, o campo status é um enum com os seguintes valores:

* SCHEDULED -> Indica que o quarto selecionado está reservado para o período de check-in escolhido
* IN\_USE -> Indica que o quarto está neste momento ocupado pela reserva realizada
* ABSENCE -> Indica que o responsável pela reserva não compareceu ao hotel para se hospedar. (Nesse cenário, este status é considerado como situação final da reserva e não deve ser atualizado para manter histórico)
* FINISHED -> Indica que a reserva foi concluída com sucesso e o cliente já deixou o hotel. (Nesse cenário, este status é considerado como situação final da reserva e não deve ser atualizado para manter histórico)
* CANCELED -> Indica uma reserva cancelada antes de iniciar o check-in. (Nesse cenário, este status é considerado como situação final da reserva e não deve ser atualizado para manter histórico)

Instruções para o desenvolvimento:

1. Crie um projeto Java utilizando Spring Boot e estruture as camadas do projeto (controller, service, repository, etc..)
2. Faça a integração com o banco de dados e mapeie as entidades no projeto. (Neste momento, a estrutura das tabelas devem ser criadas automaticamente via hibernate)
3. Crie 1 endpoint para customers e outro para rooms contendo uma operação completa de CRUD para cada um.
4. Crie 1 endpoint que permita a abertura de uma nova reserva
5. Crie 1 endpoint que permita a o encerramento de uma reserva
6. Crie 1 endpoint que retorne todas as reservas dados um intervalo de datas
7. Crie 1 endpoint que permita visualizar os quartos que estão ocupados no momento.
8. Agora, pense que você está participando de uma cerimônia de refinamento desta demanda, e o Tech Lead condutor da dinâmica questiona o time se há alguma melhoria possível de ser realizada neste modelo, você sugeriria algum ajuste?

Ah, e não se esqueça de incluir logs em todas as operações realizadas no sistema, pois caso aconteça algum bug, precisaremos entender tudo que ocorreu no processo executado.

# Etapa 2 - Avaliação de banco de dados

Neste momento, iremos avaliar seus conhecimentos de SQL com alguns questionamentos que chegaram através da nossa Product Owner. O cliente pediu algumas análises de base e precisamos compilar as informações para entregar a ele.

**OBS: Responda cada item com a query a ser executada em base para trazer as informações que precisamos.**

1. Quantos clientes temos na base?
2. Quantos quartos temos cadastrados?
3. Quantas reservas em aberto o hotel possui no momento?
4. Quantos quartos temos vagos no momento?
5. Quantos quartos temos ocupados no momento?
6. Quantas reservas futuras o hotel possui?
7. Qual o quarto mais caro do hotel?
8. Qual o quarto com maior histórico de cancelamentos?
9. Liste todos os quartos e a quantidade de clientes que já ocuparam cada um.
10. Quais são os 3 quartos que possuem um histórico maior de ocupações?
11. No próximo mês, o hotel fará uma promoção para os seus 10 clientes que possuírem maior histórico de reservas e você foi acionado pelo seu time para extrair esta informação do banco de dados. Quem são os 10 clientes?

Boa Sorte!