

Objetivo

Fazer uma API REST em **Spring Boot** persistindo os dados no **Mysql** ou **MongoDB**.

Os fontes devem ser disponibilizados em um repositório público: **Github** ou **Bitbucket**.

Se não conseguir fazer tudo, não se preocupe. Traga suas dúvidas e nos envie os fontes mesmo assim!

Contexto

Uma locadora de carros precisa de um sistema para gerenciar as locações. Ela precisa das seguintes informações:

- Nome do cliente;
- Data inicial da locação;
- Data final da locação.

Pelo sistema, o usuário deverá conseguir cadastrar, consultar e cancelar uma locação.

Restrições:

- A data inicial não deve ultrapassar a data final;
- Se o cliente já possui uma locação, não poderá fazer outra;

Tecnologias que devem ser empregadas

- API deve ser escrita utilizando java (7 ou 8) utilizando o framework Spring Boot;
- Use o framework de persistência de sua preferência.

Sugestões

- Criar testes unitários para a regra de negócio;
- Utilizar o Docker para subir o banco de dados.
 - Imagem **Docker** do banco de dados:
 - https://hub.docker.com/_/mysql/
 - https://hub.docker.com/_/mongo/

Requisitos obrigatórios

- Documentação resumida de como utilizar a API. Pode ser um arquivo exportado pelo Postman.

Funcionalidade Opcional

Fazer um endpoint no **mesmo serviço** java criado, que rode através de uma chamada POST Rest (**/calcular-dias-restantes**), que irá calcular os dias restantes da locação por cliente.

Esse endpoint deverá calcular os dias restantes por cliente e, para cada cliente, publicar uma mensagem contendo o nome do cliente e o número de dias restantes em uma queue **RabbitMQ**.

A mensagem deverá ser consumida por um consumer feito em **Python** e os dados devem ser persistidos no **mesmo banco de dados**.

- Imagem docker do RabbitMQ:

- https://hub.docker.com/_/rabbitmq/

Referências

- <https://spring.io/guides/gs/rest-service/>
- <https://www.rabbitmq.com/api-guide.html>
- <https://www.rabbitmq.com/tutorials/tutorial-two-python.html>
- <https://docs.docker.com/get-started/#test-docker-installation>
- <https://www.getpostman.com/>

