Atenção, apaixonados por tecnologia! Quer fazer parte da empresa que vai acelerar sua carreira? Estamos procurando desenvolvedores para se juntar ao nosso time.

Mas afinal, o que esperamos de você?

- Ser apaixonado(a) por tecnologia;
- Gostar da teoria e mais ainda de colocar a mão na massa;
- Estar sempre em busca de novos conhecimentos na área de tecnologia;
- Ser motivado por desafios tecnológicos ;
- Sentir prazer por melhorias contínuas;
- Ter proatividade, sentimento de dono e colaboração;
- Entusiasmo na busca de soluções;
- Aprender e absorver conhecimento rapidamente.

Se lendo esses tópicos você já curtiu nossa pegada e quer conhecer mais sobre a Dryve, acesse: https://dryve.com.br/

Aqui vão alguns requisitos mínimos pra você se tornar um Dryver:

- Experiência com linguagem de programação Java 8+ ou Kotlin;
- Experiência com Spring framework;
- Experiência em microserviços;
- Experiência com banco de dados relacional;
- Experiência no desenvolvimento de APIs Rest.

E alguns conhecimentos desejaveis:

- Conhecimento em plataformas de mensageria (Kafka ou RabbitMQ);
- Conhecimento de Arquitetura Hexagonal e boas práticas de programação;
- Conhecimento de paradigmas de DevOps;
- Conhecimento de metodologias ágeis como Kanban ou Scrum.

Curtiu? Então chegou a hora de mostrar o seu talento. Para nos conhecermos melhor, criamos um teste para você. Reserve um lugar tranquilo e adequado para implementar o teste. Depois de finalizado, envie o link do repositório para avaliarmos, beleza?

Teste

Criar um serviço de cadastro de veículos onde o usuário fornecerá os dados básicos e o serviço deverá salvar esses dados para posterior consulta. Porém antes de salvar estes dados, o serviço deverá consultar em nossa API da KBB para buscar o valor desse veículo e salvá-lo junto com os dados básicos.

O serviço deverá listar os veículos cadastrados de forma paginada e permitir a filtragem dessa lista através da placa. Os dados exibidos na listagem são id, placa, marca e modelo.

Além disso, o serviço também deverá fornecer a consulta dos detalhes a partir do id retornando todos os dados vinculados a esse registro.

O processo para cadastro dos veículos deverá seguir o seguinte fluxo:

- Serão recebidos pela aplicação os seguintes dados do veículo para cadastro:
  - **Placa**
  - **ID Marca (ex: ‘ca43ec74-5bb0-4288-ab11-5df094ca4dc4’)**
  - **ID Modelo (ex: ‘5bc16064-d3ee-4aed-a264-a914233d0c4f’)**
  - **Preço do anúncio**
  - **Ano**

- A partir dos dados recebidos, será consultado o valor na tabela KBB do veículo utilizando a API disponível em **https://6048bdf1fb5dcc0017968e3f.mockapi.io/api/v1/kbb** (ex: https://6048bdf1fb5dcc0017968e3f.mockapi.io/api/v1/kbb/prices/1)

- Todos os dados do veículo deverão ser salvos em um banco de dados relacional:
  - **Placa**
  - **Preço do anúncio**
  - **Ano**
  - **Preço da KBB**
  - **Data do Cadastro**

- Deverá existir a relação (FKs) entre marca, modelo e veículo no banco de dados;

- Não poderá ser cadastrado um veículo com uma placa que já esteja cadastrada;

- Não poderá ser cadastrado um veículo com marca, modelo e/ou ano que não existam na consulta dos dados da tabela da KBB ou não existam no banco de dados do serviço;

- Os dados do veículo no resultado da consulta deverão ser no seguinte modelo:

```
{
        "placa": "XYZ-1234",
        "preco_anuncio": 1234.56,
        "ano": 2020,
        "preco_kbb": 120000.00,
        "data_cadastro": "2020-05-18",
        "modelo": "UNO",
        "marca": "FIAT"
}
```

Crie um repositório no github e descreva o passo a passo no README para o funcionamento, juntamente com as tecnologias utilizadas e as demais informações que achar relevante.

UTILIZAR

- Spring Boot com Java ou Kotlin
- Maven ou Gradle
- Banco de dados relacional
- APIs REST para cadastro e consulta dos dados

DIFERENCIAL

- Testes de integração
- Testes unitários
- Dockerfile para container do projeto

- Docker compose para dependências do projeto
- Publicação dos eventos de cadastros utilizando RabbitMQ
- Versionamento de banco de dados (flyway ou liquibase)

Preze muito pela qualidade do código e tente não se esquecer dos testes, eles serão avaliados com muito carinho pela nossa equipe de tecnologia da Dryve.

Boa sorte e até breve!

Insert das marcas e modelos

```
INSERT INTO brand (id, name) VALUES ('ca43ec74-5bb0-4288-ab11-5df094ca4dc4', 'FIAT', 1);
INSERT INTO brand (id, name) VALUES ('c0225595-e293-477b-8758-384872470f00', 'FORD', 2);

INSERT INTO model (id, name, brand_id) VALUES ('5bc16064-d3ee-4aed-a264-a914233d0c4f', 'TORO 2.0
DIESEL', 'ca43ec74-5bb0-4288-ab11-5df094ca4dc4');
INSERT INTO model (id, name, brand_id) VALUES ('e16e9ed4-43c6-4882-9f2f-12ca5aad6e7e', 'ARGO 1.0
Flex.', 'ca43ec74-5bb0-4288-ab11-5df094ca4dc4');
INSERT INTO model (id, name, brand_id) VALUES ('c08f77df-c6e0-4e73-a378-42bb83361266', 'Onix 1.0
Turbo Flex', 'c0225595-e293-477b-8758-384872470f00');
INSERT INTO model (id, name, brand_id) VALUES ('deaf80e7-600c-4a35-af52-1fc7f1e967a8', 'EcoSport
1.6 Flex', 'c0225595-e293-477b-8758-384872470f00');

INSERT INTO model_year (id, model_id, year, kbb_id) VALUES ('9d31e563-9d09-4ce8-8ab5-
c5177f51d92f', '5bc16064-d3ee-4aed-a264-a914233d0c4f', 2020, 1);
INSERT INTO model_year (id, model_id, year, kbb_id) VALUES ('b9824542-0d28-4e09-92df-
53379e8b3b22', '5bc16064-d3ee-4aed-a264-a914233d0c4f', 2021, 2);
INSERT INTO model_year (id, model_id, year, kbb_id) VALUES ('f78e6d41-c620-4a76-a0a4-
eb19f45c623e', 'e16e9ed4-43c6-4882-9f2f-12ca5aad6e7e', 2020, 3);
INSERT INTO model_year (id, model_id, year, kbb_id) VALUES ('609c2dbf-c563-486a-8660-f4a17fe63818',
'e16e9ed4-43c6-4882-9f2f-12ca5aad6e7e', 2021, 4);
INSERT INTO model_year (id, model_id, year, kbb_id) VALUES ('88748742-ed8c-467c-8990-
963bfd617a0e', 'c08f77df-c6e0-4e73-a378-42bb83361266', 2020, 5);
INSERT INTO model_year (id, model_id, year, kbb_id) VALUES ('d9f4ec9e-35db-4ec2-9cab-
1bf6a0011557', 'c08f77df-c6e0-4e73-a378-42bb83361266', 2021, 6);
INSERT INTO model_year (id, model_id, year, kbb_id) VALUES ('2dc1afe8-7851-4edc-9b91-
a507c692483b', 'deaf80e7-600c-4a35-af52-1fc7f1e967a8', 2020, 7);
INSERT INTO model_year (id, model_id, year, kbb_id) VALUES ('6aa4d860-f5a8-439a-ab50-
fda16b45549c', 'deaf80e7-600c-4a35-af52-1fc7f1e967a8', 2021, 8);
```