

DESAFIO BACKEND

É um prazer ter você no nosso processo de seleção.

Aqui vai a especificação de um desafio simples de backend para poder termos uma ideia das suas habilidades.

Observação importante

A intenção aqui é criarmos um projeto que evolui gradualmente em etapas. **Não é obrigatória a realização de todas as etapas. No entanto, cada etapa concluída do desafio pode nos ajudar a identificar e qual nível técnico você melhor se adequa.**

Contexto

Uma entidade Rota é composta de uma data, um status que pode conter os valores NOT_STARTED, STARTED e DONE e paradas. Cada parada contém uma descrição, um endereço com latitude e longitude e status que pode ser NOT_ANSWER (para não atendido) e ANSWER (para atendido). A seguir um exemplo de rota válida:

```
{
  "id": 1,
  "date": "2021-10-14",
  "status": "NOT_STARTED",
  "stops": [
    {
      "description": "X do João",
      "address": "Av. Washington Soares, 723 - Edson Queiroz,
Fortaleza - CE, 60811-341",
      "latitude": -3.7625982,
      "longitude": -38.4841062,
      "status": "NOT_ANSWER",
      "deliveryRadius": 10
    },
    {
      "description": "Galeto Prime - Delivery iFood",
      "address": "Rua Professor Eládio Magalhães, 213 - Edson
Queiroz, Fortaleza - CE, 60811-460",
      "latitude": -3.751316,
      "longitude": -38.51507,
      "status": "NOT_ANSWER",
      "deliveryRadius": 30
    }
  ]
}
```

Etapa 1 - A aplicação básica

Crie um micro serviço contendo uma API REST que permita realizar as operações de CRUD (Create, Recover, Update and Delete) de Rotas, obedecendo corretamente os verbos e status codes correspondentes e respeitando as seguintes regras:

- Não será necessário implementar quaisquer autenticações na API.
 - Para criar uma rota, deve-se enviar uma requisição POST para /Route contendo uma rota válida.
 - Uma rota deve possuir ao menos uma parada.
 - Não deve ser permitido adicionar paradas que possuam coordenadas inválidas (latitude e ou longitude igual a zero ou null).
 - Consistência eventual não será um problema.
 - Pode utilizar qualquer base de dados e qualquer solução de cache.
 - A exposição direta de entidades na sua API não será permitida.
-

Etapa 2 - Log de eventos

Sua aplicação pode ainda possuir um sistema de logging onde a cada mudança de status na rota, um novo evento é registrado contendo: id, data do evento e status da rota. O sistema de log pode ser escrito em um outro micro serviço que recebe de forma assíncrona os eventos de alteração de status das rotas.

Publishers/Subscribers podem ser utilizados para desenhar a solução.

Etapa 3 - Eventos de atendimento

Uma terceira opção é atualizar o status das paradas recebendo coordenadas. Uma coordenada representa a posição de veículo em um determinado instante. Para ilustrar, você poderia imaginar uma coordenada no seguinte formato:

```
{
  "lat": -3.734652,
  "lng": -38.469755,
  "instant": "2021-08-08T23:48:15+00:00",
  "routeId": 1
}
```

Sua aplicação deve considerar que uma parada teve seu atendimento iniciado se ela consumir duas coordenadas consecutivas dentro do raio de entrega da parada (em metros), desde que o intervalo entre as duas tenha sido de pelo menos 5 minutos. E deve considerar que uma parada teve seu atendimento encerrado ao consumir a primeira coordenada fora do raio de entrega da parada após o início de atendimento.

Sua aplicação deve responder às seguintes perguntas, criando um endpoint para cada uma:

- Num dado instante, quais paradas foram atendidas para cada rota?
- Num dado instante, qual é o status de cada rota (Pendente, Em progresso, Concluída)?
- Quais foram as paradas mais demoradas para cada rota?

As atualizações dos status das paradas também podem ser enviadas para seu sistema de logging.

Etapa Bônus

Responder ou implementar soluções que respondam as seguintes perguntas:

- Uma vez que você realizou ao menos a etapa 1, como sua aplicação vai para produção (deployment)?
- Se uma falha ocorrer, como podemos visualizar os logs da sua aplicação em produção?
- Que métricas você considera importantes existirem na sua aplicação.
- Caso eu não conheça seu projeto, como posso fazê-lo funcionar sem ter que te perguntar pessoalmente?
- Se uma mudança no comportamento da sua API atual for realizada, os seus testes atuais ainda vão continuar funcionando?
- Qual a carga máxima de sessões e conexões que sua aplicação suporta? justifique.

Considerações finais

Sinta-se livre para criar seus próprios planos de rota e modelar suas entidades usando quaisquer atributos que você julgar necessário. Contamos com a sua criatividade!

Você será responsável tanto pela emissão de coordenadas quanto pelo recebimento de coordenadas, sugerimos microserviços separados, mas você está no controle!

Estamos interessados em ver como você preencherá os pontos vagos desse desafio, mas não hesite em tirar dúvidas quando sentir que for necessário.

Dicas:

Nós gostamos muito de arquiteturas orientadas a eventos, você pode pesquisar sobre event-sourcing e sobre CQRS.

Testes são muito importantes pra gente!

Não ter frontend não afetará de forma negativa na avaliação, mas se quiser fazer, fique à vontade!

Aqui na Descartes trabalhamos, principalmente, com: - Microserviços; - Java 8, 11 e Kotlin; - Spring Boot / Spring Cloud; - Docker, Kafka, Redis, MongoDB e Oracle.