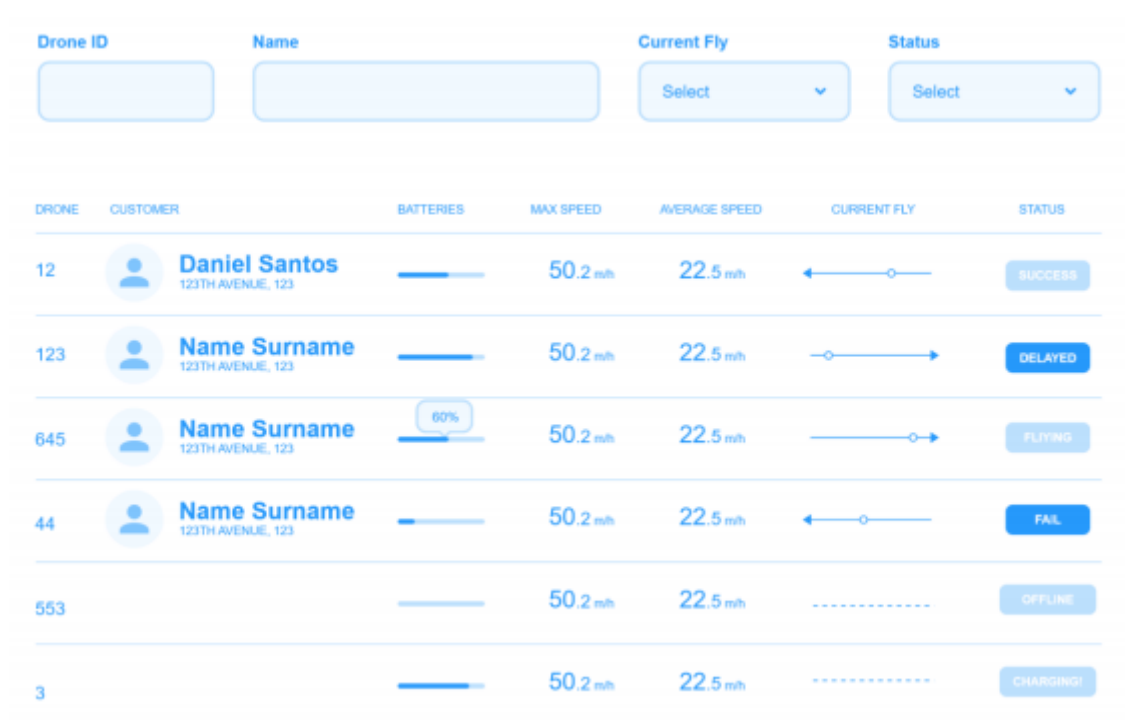


Desafio técnico | Back end

(v19.06)

O objetivo da avaliação é criar uma back-end de dados de entregas por drones. O mockup da tela é da seguinte forma. Obs: o front-end não está incluso no teste.



Drone ID	Name	Current Fly	Status
		Select	Select





DRONE	CUSTOMER	BATTERIES	MAX SPEED	AVERAGE SPEED	CURRENT FLY	STATUS
12	 Daniel Santos 123TH AVENUE, 123	<div><div></div></div>	50.2 m/h	22.5 m/h	<div><div></div></div>	SUCCESS
123	 Name Surname 123TH AVENUE, 123	<div><div></div></div>	50.2 m/h	22.5 m/h	<div><div></div></div>	DELAYED
645	 Name Surname 123TH AVENUE, 123	<div><div>60%</div></div>	50.2 m/h	22.5 m/h	<div><div></div></div>	FLYING
44	 Name Surname 123TH AVENUE, 123	<div><div></div></div>	50.2 m/h	22.5 m/h	<div><div></div></div>	FAIL
553		<div><div></div></div>	50.2 m/h	22.5 m/h	<div><div></div></div>	OFFLINE
3		<div><div></div></div>	50.2 m/h	22.5 m/h	<div><div></div></div>	CHARGING

Figura 1 - Mockup de layout de Drones e suas Entregas

Aplicação

Você deve fazer uma aplicação Back-end REST API para fornecer os dados para este front-end. Os seguintes recursos e métodos são necessários:

```
List: GET /drones Detail: GET  
/drones/{id} Update: PUT /drones/{id}  
Delete: DEL /drones/{id} Create: POST  
/drones/{id}
```

```
Paginate: GET /drones?_page=7&_limit=20 Sort: GET  
/drones?_sort=id&_order=asc  
O Recurso "Drone" deve ser da seguinte forma:
```

```
{  
  "id": 1, "customer_image": "https://robohash.org/uia.jpg",  
  "customer_name": "Suzann", "customer_address": "955 Springview  
Junction", "battery": 90, "max_speed": 3.8, "average_speed": 11.6,  
  "status": "failed", "current_fly": 94 }
```

Requisitos Obrigatórios

- A aplicação deve ser Nodejs.
- Banco de dados SQLite.
- Deve utilizar algum Framework REST de sua preferência.
- Deve utilizar algum ORM de sua preferência.
- Utilizar Typescript (opcional).
- A API deve fazer validação dos dados, Todos os campos, exceto customer_* são obrigatórios
- Deve se utilizar controle de versão GIT.
- Read-me com instruções para executar API (npm install, npm run start, etc...)

O que vamos avaliar

- Adequação aos requisitos mencionados acima.
- Erros e bugs na aplicação.
- Documentação do código
- Organização e legibilidade do código.
- Lógica das soluções apresentadas.
- Adequação à padrões de codificação.
- Coesão e acoplamento.

Boa sorte! (: