

## Programa IT Academy – Processo Seletivo – Edição #18

Nome Completo: Gabriel Auth Maldaner Griebeler

E-mail: gabrieltgriebeler98@gmail.com

### Etapa 1 – Questões de lógica

Esta seleção possui 15 questões de lógica de caráter eliminatório. As questões são apresentadas no formulário de Exercício Técnico e devem ser respondidas no próprio formulário online, que deverá ser acessado através do link a seguir: <https://forms.gle/yZtVcv1b5fCgScLBA>

### Etapa 2

### RESUMO DA SOLUÇÃO

Para cumprir a demanda solicitada foi pensando em fazer um sistema mobile utilizando React Native (por ser multiplataforma) e uma api spring boot para poder ser consumida.

Para rodar a aplicação é necessário subir o servidor de API e também o servidor mobile onde se encontra o frontend.

O app possui quatro telas, sendo a principal que possui quatro botões, um para cada funcionalidade solicitada, e, cada uma das funcionalidades (tirando a funcionalidade de finalizar aplicação) possuem uma tela própria.

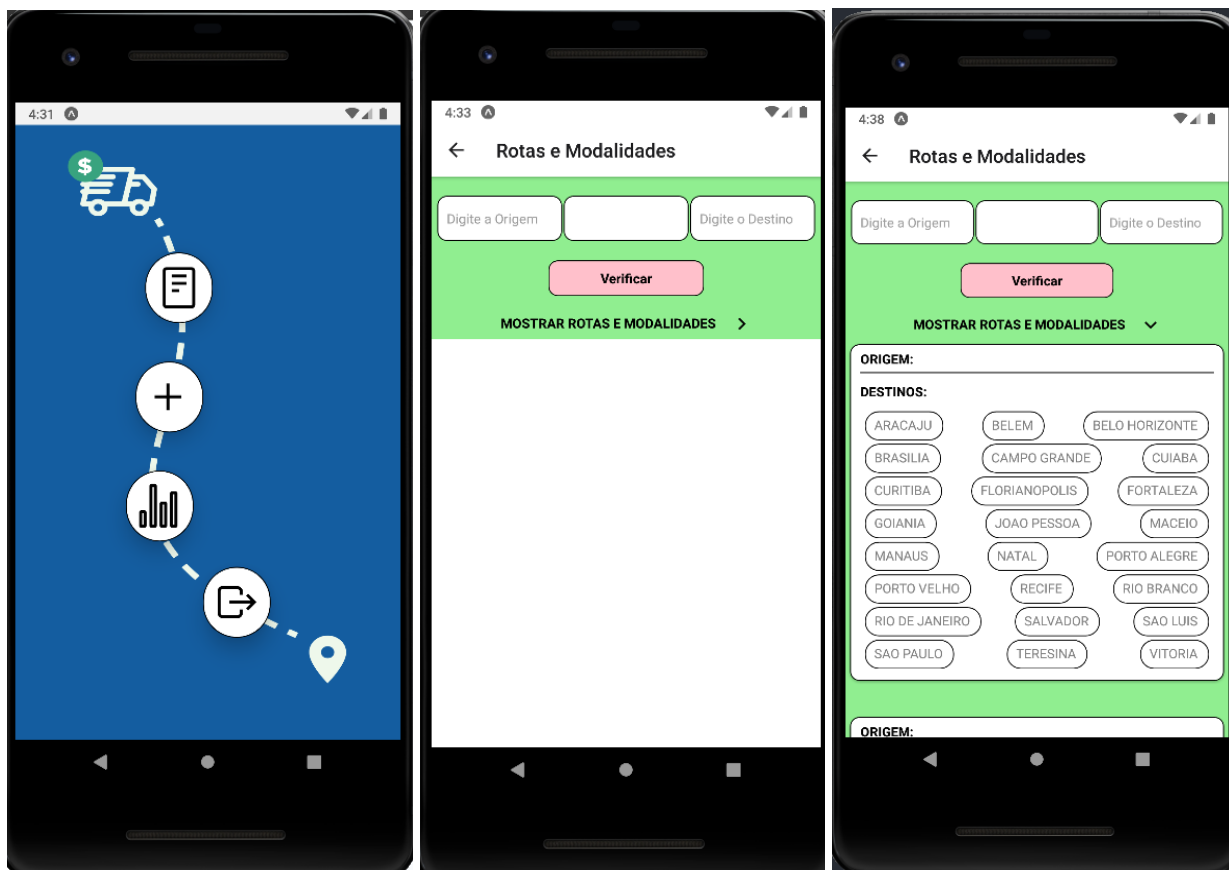
Cada tela ao iniciar faz a requisição conforme a demanda solicitada para a api, essa por sua vez por utilizar do Spring Boot recebe a requisição em um Controller (CargoFeeSimulatorController) através do protocolo http, que por sua vez chama o arquivo que contém o Service (CargoFeeSimulatorService) que processa os dados e retorna para a aplicação.

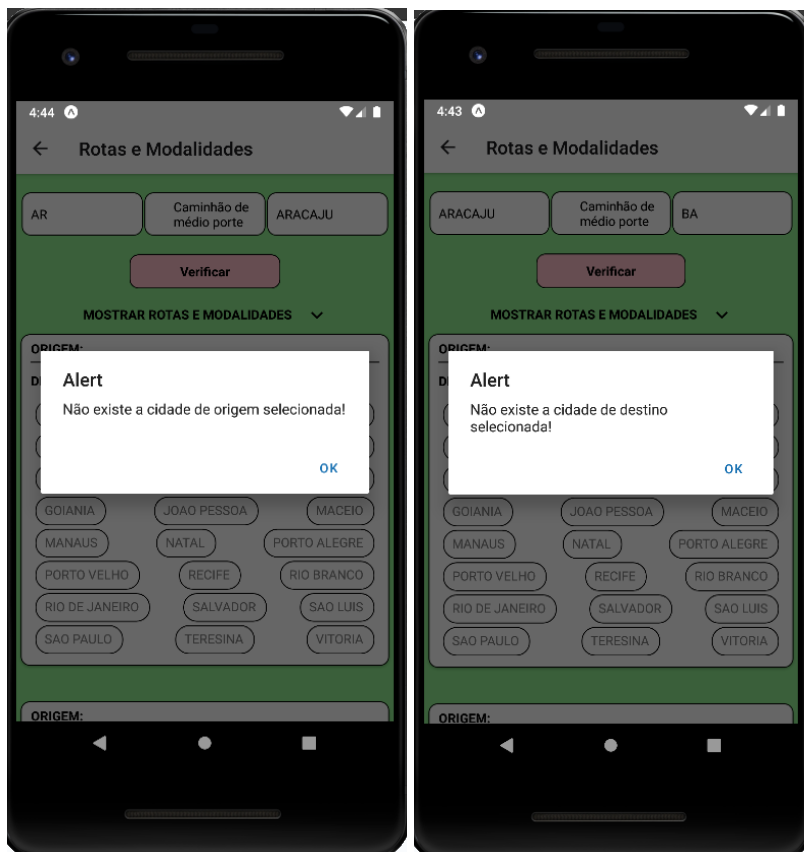
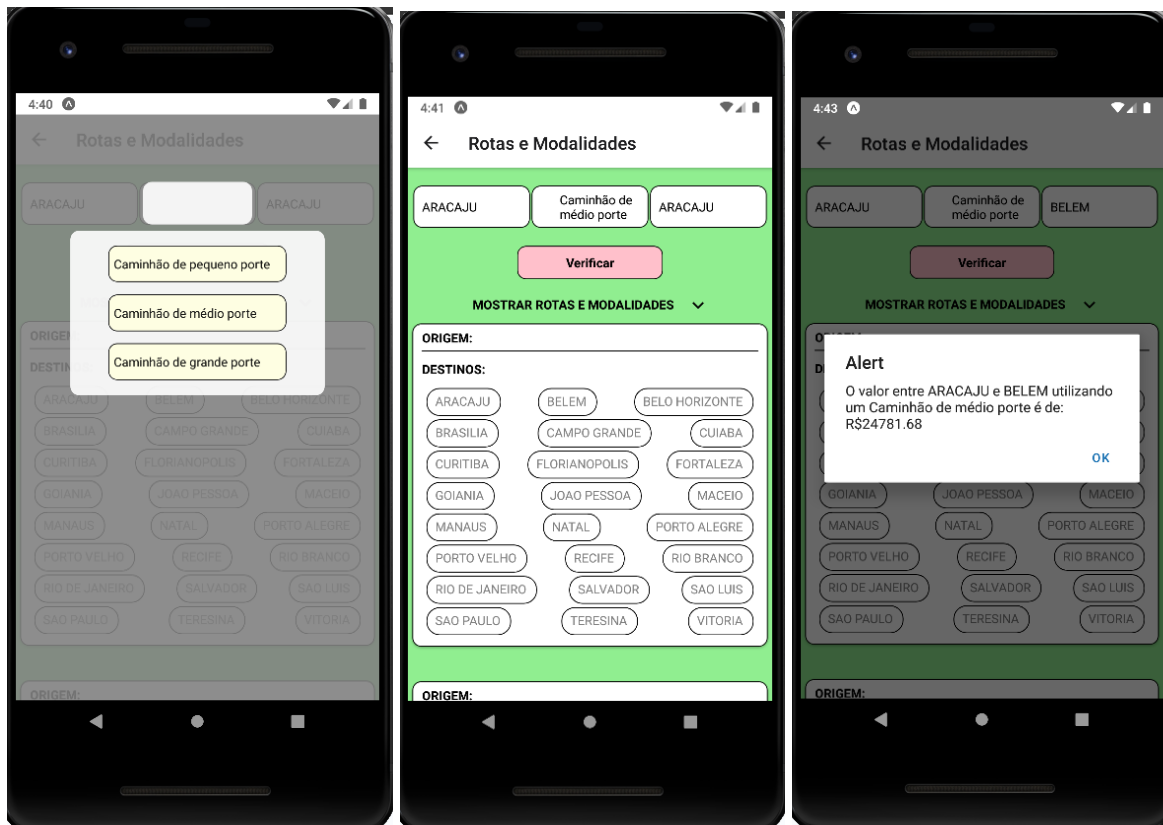
Com relação ao backend, o arquivo é lido e guardado em memória em um atributo da classe FileService.java na inicialização da API. Em cada requisição, esse atributo é acessado e a partir dele é montado um Array onde cada index é uma cidade de origem, e esta, possui seu nome e um Array que contém o nome do destino, ou seja, todas as outras cidades.

O encapsulamento de cada requisição depende de cada funcionalidade será explicado posteriormente na sua devida seção, junto com os prints relacionados.

**TESTES** (aqui você deverá colar capturas de tela de todas as funcionalidades desenvolvidas e realizar comentários, use o espaço que julgar necessário)

### **FUNCIONALIDADE 1 – Consultar trechos x modalidade**



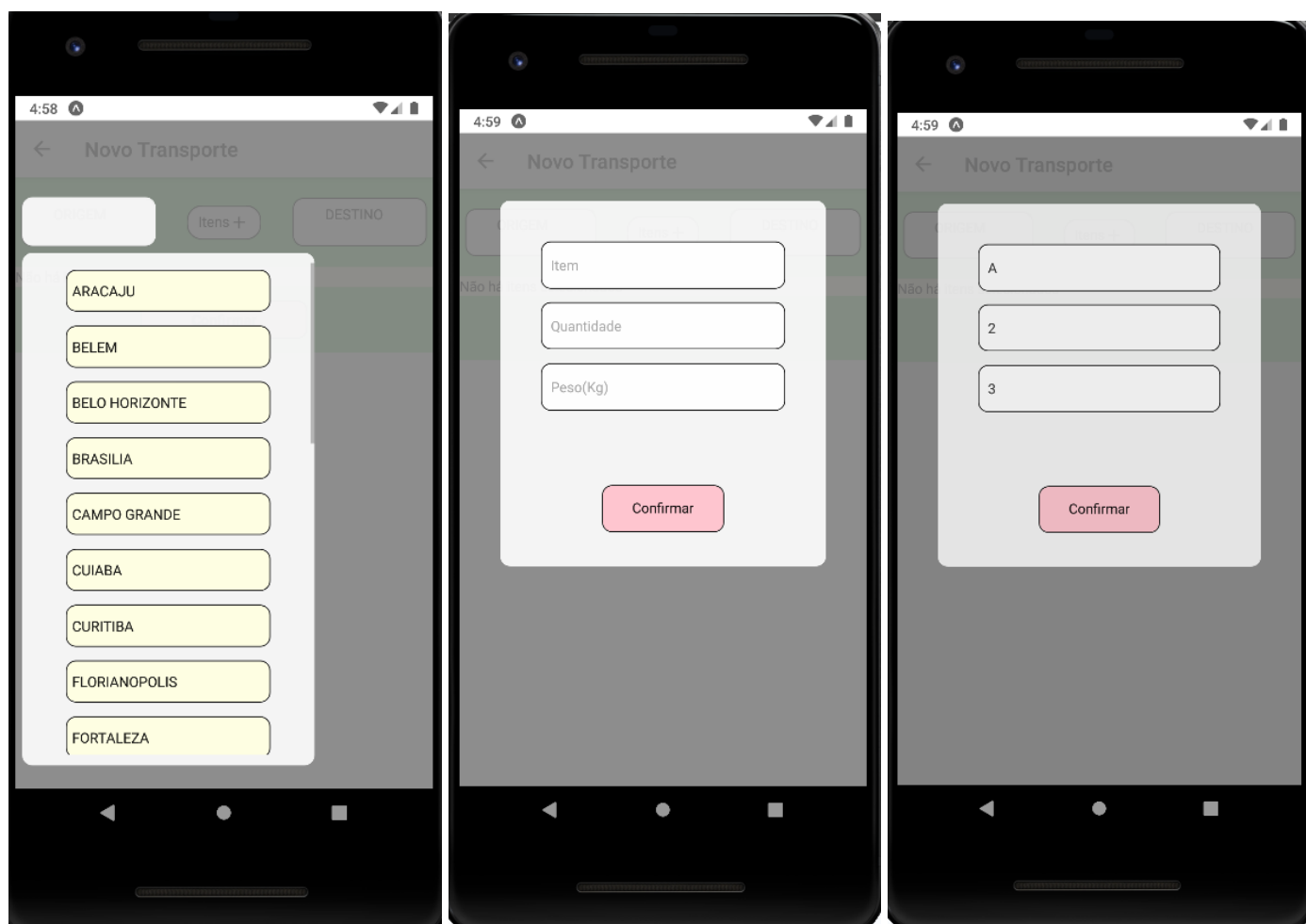


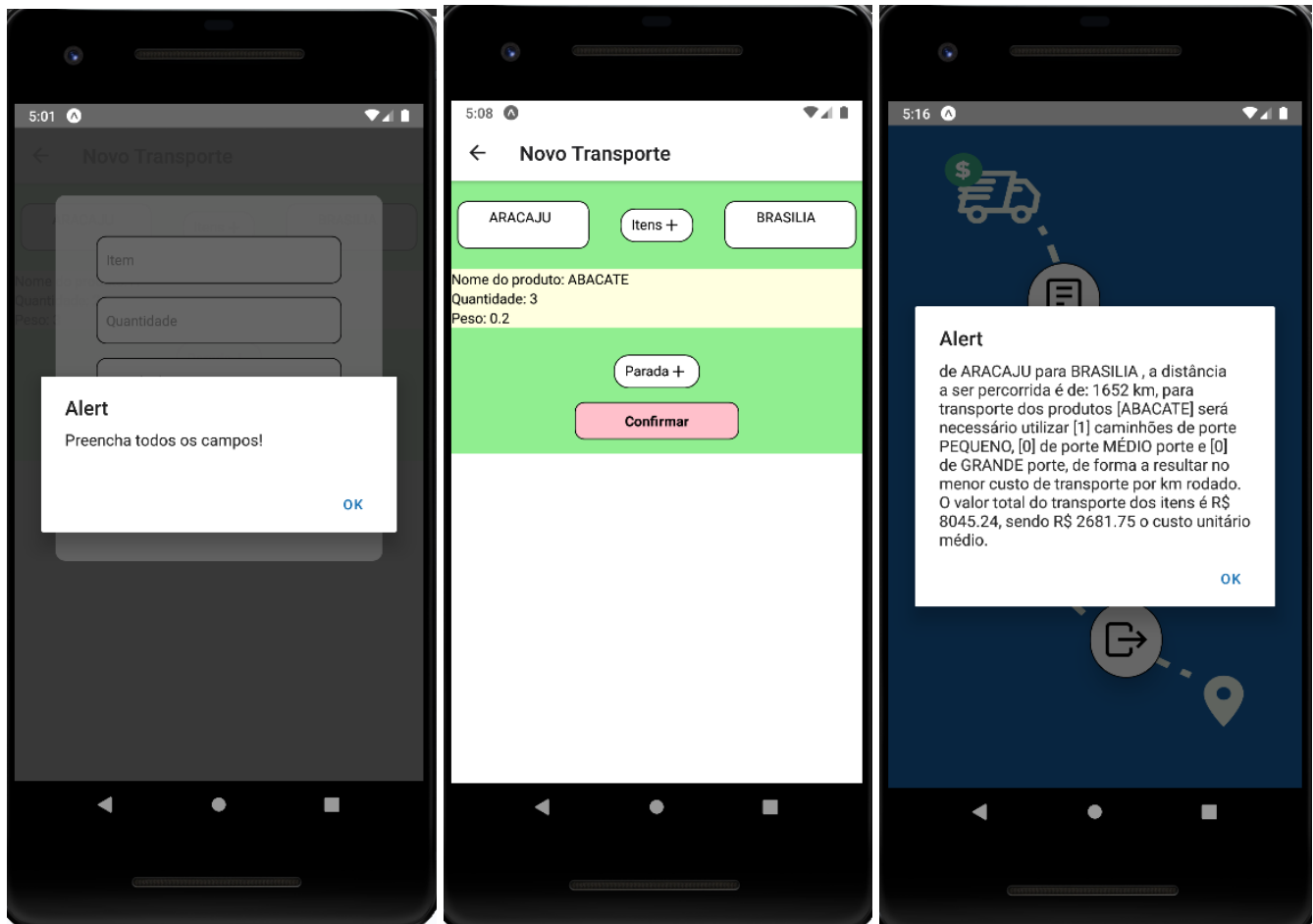
Para essa requisição foi pensado em ao iniciar a tela ele fazer uma requisição que retorna para o front o array com todos os nomes de cidade, que ao clicar em MOSTRAR ROTAS E MODALIDADES é iterado cada cidade origem para todas as mesmas cidades agora listadas como destino.

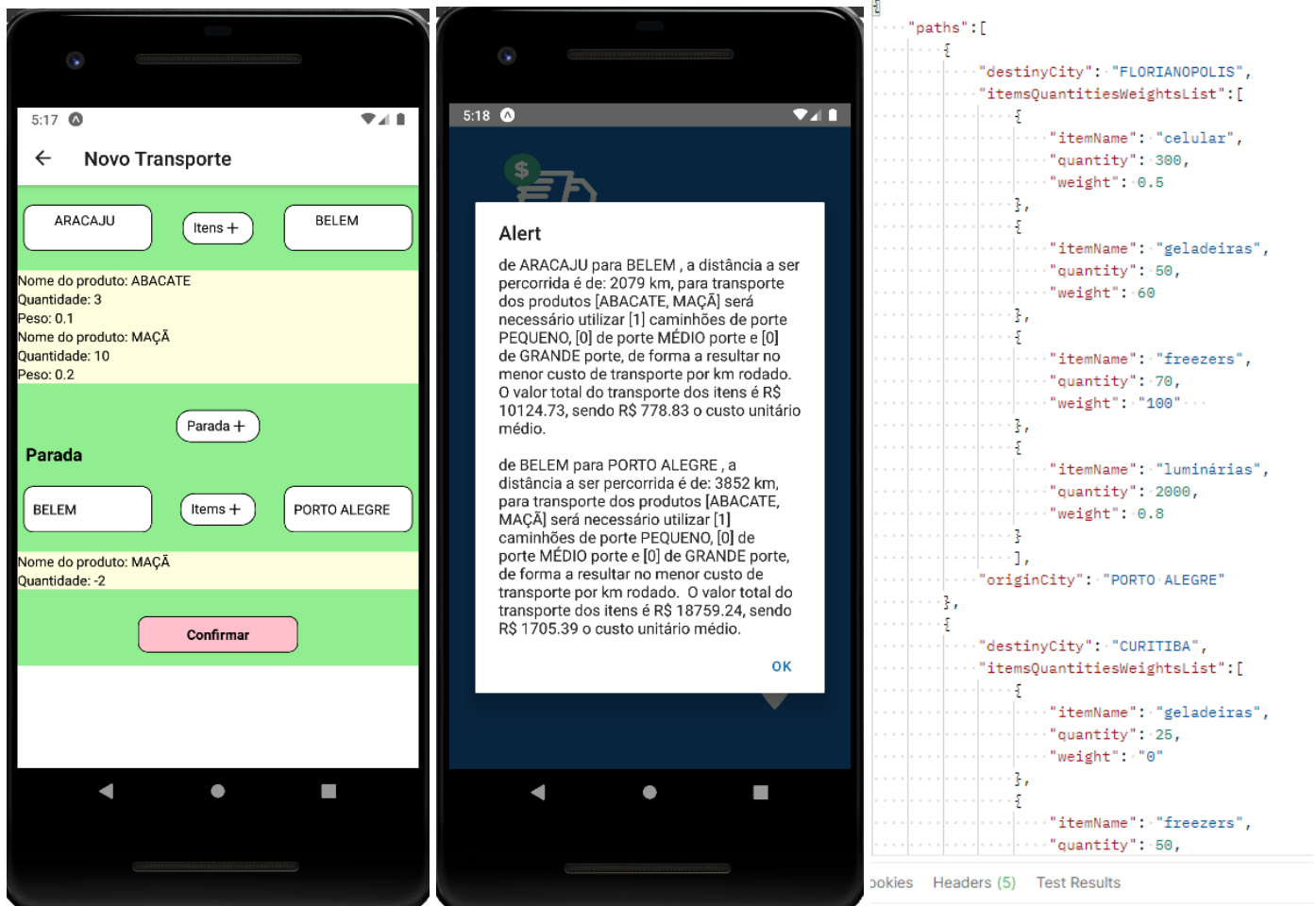
Para a verificação de transporte, é passado o nome da cidade de origem e de destino bem como a modalidade de transporte que é estática. Essa requisição tem como retorno, caso as duas cidades existam o valor da distância, caso contrário ele verifica qual dos parâmetros foi passado erradamente e retorna o erro.

Por trás do backend temos um filter, para cada objeto do Array lido do CSV que verifica e retorna o valor do array que contém a cidade de origem e novamente caso validade é feito outro filter que retorna o valor que bate com com o destino passado pela request, após é pega a distância e multiplicada pelo valor estático da modalidade de transporte, e é retornada uma mensagem para o usuário.

## FUNCIONALIDADE 2 – Cadastrar transporte







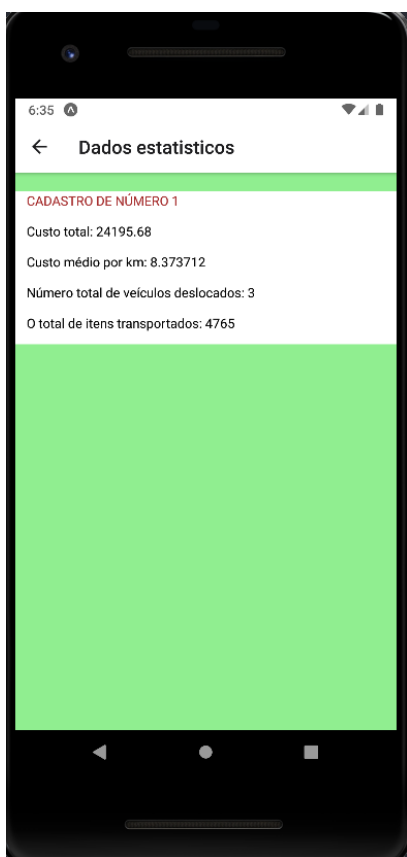
Para essa funcionalidade foi pensado em no front foi necessário encapsular em um objeto com o nome “Paths” este paths corresponde aos trajetos/caminhos possível de se cadastrar, sendo ele uma ida direta até o ponto final, ao ainda, com uma parada para descarga de itens, sendo assim, esse path é um array com dois objetos.

Cada um dos objetos carrega consigo sua cidade de origem, cidade de destino e ainda uma lista com cada item a ser transferido bem como seu peso e sua quantidade. Caso seja necessário uma parada, ao informar ele preenche automaticamente o a cidade de origem da para, já que afinal, é uma parada e permite que o usuário informe quais produtos ele gostaria de descarregar bem como a cidade de destino(final). E, para o caso de não ser necessária a parada, ele salva o Path com somente um objeto(Rota) e envia para o backend, validando a maioria dos preenchimentos dos campos.

Com relação ao backend é feito o desempacotamento do JSON que vem do frontend através de classes que definem o que é cada um destes itens, instanciando

elas para posteriormente utilizar para os cálculos, comparações e ainda, armazenando os dados para posteriormente ler nos dados estatísticos. O último print é relacionado a um exemplo do objeto a ser enviado pela requisição.

### FUNCIONALIDADE 3 – Dados estatísticos



Ao acessar a tela de estatísticas ele faz a requisição para o backend solicitando os dados, que por sua vez retorna um array com quase todos os dados solicitados e que por sua vez, percorre o array pega cada instância do dado e printa na tela.

Conforme dito na funcionalidade anterior, os dados são persistidos em variáveis que são aproveitadas para montar o objeto Cargo que possui como atributos os valores para preenchimento das estatísticas, cada requisição para Criar Transporte ao final cria um objeto do tipo Cargo e salva num array(chamado de cargos) do tipo Cargo que

guarda o histórico de dados já cadastrados ao receber a requisição de get do usuário pelo front, ele faz o get do Array cargos, do tipo Cargo.

## AUTOAVALIAÇÃO

Você concluiu a implementação de 100% das funcionalidades solicitadas?

( ) Sim      (X) Não

Para as 3 principais funcionalidades solicitadas, como você avalia a sua solução?

Marque um 'X'.

	Inexistente/ Insuficiente	Pouco satisfeito(a)	Satisfeito(a)	Muito satisfeito(a)
Funcionalidade 1			X	
Funcionalidade 2			X	
Funcionalidade 3			X	

## Principais dificuldades

As principais dificuldades foram:

- como encapsular e desencapsular os dados para o envio do front para o back do back para o front e ainda, do CSV para o backend
- persistir os dados para o uso posterior.
- como processar os dados para gerar as demandas solicitadas, visto que algumas demandas solicitaram dados específicos passando por fórmulas as quais tive que rever algumas vezes.



## Desempenho Geral

Tendo em vista o tamanho do projeto, e que havia posto uma meta grande, fico um pouco triste por não ter entregue um front end bem melhor e ainda, por faltar alguns dos dados solicitados nas estatísticas, Foi utilizado Spring boot e React native.

Obrigado por participar deste processo seletivo.  
Salve o documento em PDF com o seu nome completo.