

Passar o vetor de coordenadas recebido da API no algoritmo já implementado

O que muda: algoritmo é
chamado 2x, uma com o vector
como eixo e outra com o vector invertido.
Concatena todos os pontos e exclui os
repetidos. Com isso, faz a validação **2 pass**

Passar o mesmo vetor, porém invertido (cuidado para não inverter latitude com longitude) no algoritmo já implementado.

2º chamada:
retor unperitudo

$B \rightarrow A$

Usar função truncate (turfe.js) para precisar as casas decimais das coordenadas de cada posição do vetor.

```
var truncated = turf.truncate(point, {precision: 6, coordinates: 2});  
//truncated.geometry.coordinates
```

Usar função para excluir coordenadas duplicadas consecutivas do vetor

Pesquisar cada ponto na tabela de pontos e verificar se ele pertence a alguma rota

| denton | nota |
|--------|-------------|
| a | nota x, ... |
| b | nota x |
| cw | ... |
| id | ... |
| km | ... |
| F | nota x, ... |
| y | ... |

→ entrada

saída

6 passo: Buscar os points (coordenadas) de cada Rota encontrada na Tabela de Pontos (que serão as rotas que apresentam pontos em comum com a rota a ser validada). Poderá haver mais de 1 rota em conflito. Fazer o cálculo para TODAS AS ROTAS.

7 passo: Chamar a função lineOverlap passando a rota a ser validada e a rota retirada da tabela de pontos.

```
var line1 = turf.lineString([[115, -35], [125, -30], [135, -30], [145, -35]]); //rota x ou rotaConflitante
```

```
var line2 = turf.lineString([[115, -25], [125, -30], [135, -30], [145, -25]]); // rota a ser validada
```

```
var overlapping = turf.lineOverlap(line1, line2, {tolerance: 0.005});
```

8 passo: Calcular distância entre coordenadas que foram retornadas pelo lineOverlap

9 passo: Invalidar a rota a ser cadastrada caso a distância seja maior que um valor pré estabelecido

10 passo: Exibir as várias retornadas pelo código no mapa caso a rota é dada como inválida.
Exibir mensagem de Cadastrado com Sucesso caso não seja encontrado nenhum trecho conflitante em nenhuma rota buscada na tabela de pontos
(variável trechosComOverlap retorna null para todas as rotasConflitantes pesquisadas.)