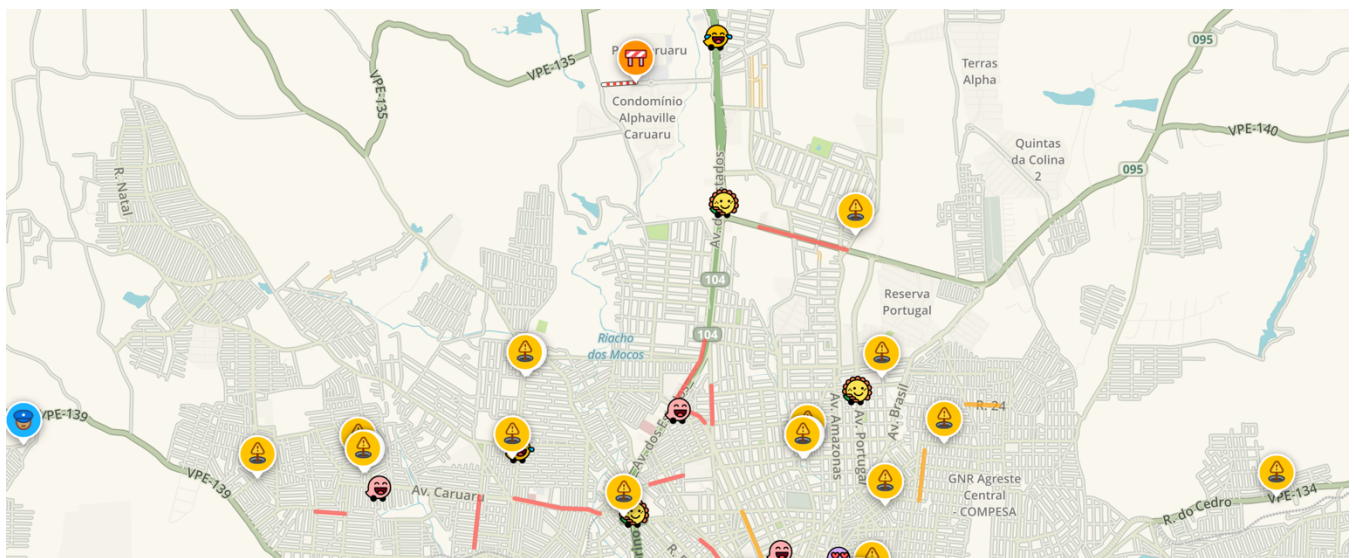
	Exercício Escolar 2	Semestre: 2025.2	
	Curso: Sistemas de Informação	Disciplina: Programação 1	
	Prof: Raphael Dourado		Nota: __/10
	Prazo de entrega: 12/11/2025		
Orientações gerais			
<ul style="list-style-type: none">• O projeto deve ser feito em equipes de 3 pessoas• No dia da entrega, cada um dos membros deve apresentar para o professor e turma a implementação do requisito que ficou responsável.			

Exercício Escolar 2 – Projeto: ueizi



Contexto: Você está desenvolvendo um app para alertar motoristas sobre perigos em rodovias brasileiras. A equipe de front-end já desenvolveu a parte do app que permite aos motoristas, colaborativamente, enviar alertas sobre diferentes tipos de perigos que encontram nas estradas. Agora é sua vez, equipe de back-end, de processar esses dados e gerar relatórios consolidados dos perigos reportados em cada rodovia.

Os alertas enviados pelos motoristas chegarão para você como arquivos CSV no formato abaixo, onde cada linha representa um alerta enviado por um usuário:

```
idUsuario;timestamp;br;km;tipoAlerta
12;1761610320;101;10.5;A
15;1761610330;232;121.3;C
111;1761610340;101;10.9;A
456;1761610345;242;91.5;B
106;1761610355;232;121.1;C
986;1761610361;242;91.9;B
```

O significado de cada coluna é descrito abaixo:

- idUsuario:** ID do usuário que reportou o perigo
- timestamp:** instante em que o perigo foi reportado, no formato epoch time (<https://www.epochconverter.com>)
- br:** código da rodovia (apenas números, 3 dígitos)
- km:** localização da do perigo na rodovia, representado como a distância em Km relativa ao início da rodovia (km 0), onde o perigo foi registrado. Esta distância é representada com 1 casa decimal.
- tipoAlerta:** os alertas podem ser de 6 tipos, representados por um dos caracteres abaixo:

A. Buraco	D. Engarrafamento
B. Via alagada	E. Via interdita
C. Acidente	F. Obras na via

Considerando este cenário, seu sistema deve implementar os requisitos abaixo:

1. **[4 pontos] Carregar e processar os arquivos CSV** com os dados de alertas enviados pelos usuários, armazenando os dados consolidados em arquivo(s), de modo que ao fechar e reiniciar o programa, todos os dados processados anteriormente estejam disponíveis para consulta e geração de relatórios. O processamento dos alertas deve seguir as regras abaixo:
 - a. Os alertas consolidados devem ser associados a trechos de 500m na rodovia. Ou seja, caso haja dois alertas do mesmo tipo, um para o km 10.3 e outro para km 10.4, eles devem ser consolidados em um único alerta para o km 10. O agrupamento deve seguir a regra:
 - i. de X.0 a X.49 = X
 - ii. de X.5 a X.99 = X.5
 - b. Sempre que dois ou mais alertas forem agrupados, deve ser armazenado o número de alertas agrupados, que será utilizado como uma métrica de “upvote”.
 - c. Note que dois ou mais arquivos diferentes de dados podem ter alertas relativos a um mesmo km de uma rodovia, de forma que as cargas têm que ser cumulativas. Exemplo: foi feita carga do arquivo 1, onde há alerta tipo C para km 10.2 da BR 232, e depois do arquivo 2, que possui alerta tipo C para km 10.4 da BR 232. Um único alerta do tipo C para BR 232 km 10 deve ser gerado, com 2 *upvotes*.
2. **[2,5 pontos] Relatório de alertas por trecho de BR:** o sistema deve permitir gerar, para uma BR informada pelo usuário e trecho dessa BR (ex.: km 100 a 150), um relatório em formato CSV (alertas_por_br.csv) com todos os alertas **consolidados** para aquela BR, apenas naquele trecho, incluindo os *upvotes*, no formato abaixo (exemplo para br 232, trecho do km 100-170):

```
br;km;tipoAlerta;upvotes
232;150.5;A;1
232;170.0;C;3
```

3. **[2,5 pontos] Relatório de número de alertas por tipo em cada BR da base:** o sistema deve permitir gerar um relatório em CSV (alertas_br_todas.csv) com o total de alertas de cada tipo para cada BR cadastrada na base, no formato abaixo:

```
br;tipoAlerta;quantidade
232;A;15
232;A;15
232;C;25
101;A;20
101;E;40
```

4. **Extra: [+1 ponto] Representação visual de alertas para uma BR:** prover no sistema uma opção para desenhar, a partir dos dados, uma representação visual da distribuição dos alertas ao longo de uma BR informada pelo usuário, utilizando um símbolo ou letra para representar cada alerta, como no exemplo abaixo:



Requisitos e regras adicionais:

- **[1.0 ponto]** O programa deve estar devidamente modularizado em funções, evitando a duplicação de código e o uso de variáveis globais.
- Os arquivos de carga CSV contêm um **número indeterminado de linhas**. O uso de listas dinâmicas para processamento dos dados é **obrigatório**. O uso de *hash tables* também é uma opção, porém não obrigatório.
- Não é permitido usar vetores ou algoritmos de ordenação durante o processo de carga e processamento. Apenas na geração dos relatórios é permitido o uso de vetores, mas evitando desperdício de memória. Exemplo: se você alocar um vetor de 1000 posições sem ter certeza de que todas as posições serão utilizadas, será penalizado na nota.
- As entradas dos usuários devem ser validadas.