



## Projeto: Processamento de arquivos de GPS

- **Automatic Vehicle System:** utilizam um módulo GPS para coleta de dados e um módulo GPRS para transmissão das informações.

- Vantagens do AVS:

- aderência aos horários;
- economia de combustível;
- automatizar a coleta de dados sobre distância e duração.

(GILLEN; CHANG; JOHNSON, 2000)

- É possível implementar o AVS em um *smartphone* para coletar os dados do GPS e armazenar em arquivo local ou base de dados na internet.

(DOLAZZI; SCALCO, 2014)



Latitude (°)	Longitude (°)	Altitude (m)	Tempo (s)
-24,177337	-47,024178	98	0
-24,177353	-47,024119	97	1
-24,177335	-47,023955	105	2
-24,177401	-47,023985	118	5
-24,177423	-47,023989	121	6

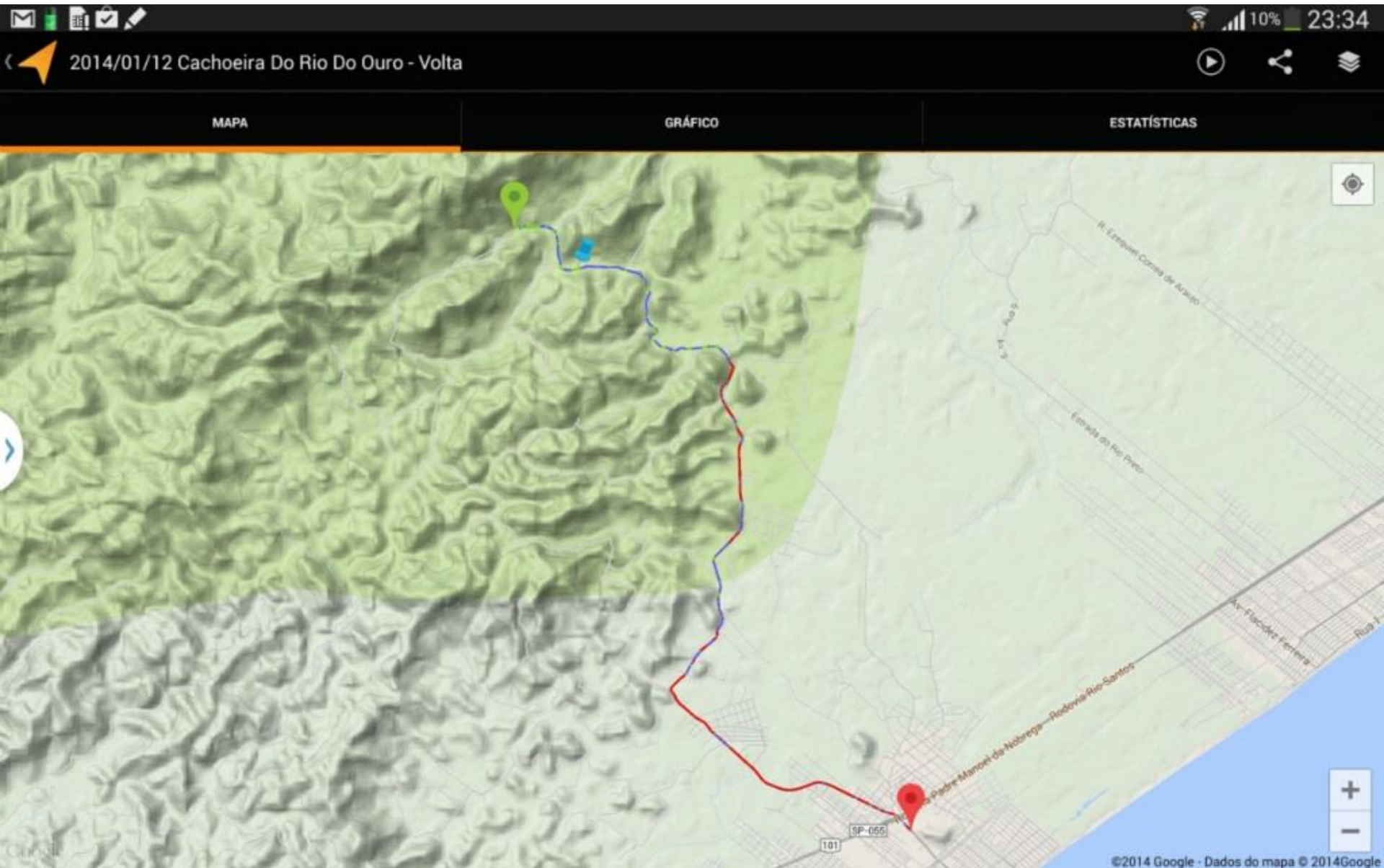
Seu programa deve:

- ler os dados do arquivo de entrada CSV;
- calcular e exibir a distância total percorrida pelo caminhão, em km;
- criar um arquivo com os dados lidos do arquivo de entrada, acrescidos da velocidade instantânea, em km/h.

No Excel:

- traçar um diagrama de dispersão com o caminho percorrido (latitude x longitude);
- traçar o gráfico velocidade x tempo, com uma linha horizontal indicando a velocidade limite de 60 km/h.

# Projeto: Processamento de arquivos de GPS



Serão lidas linhas do arquivo de entrada contendo as seguintes informações, na forma de texto, além do cabeçalho:

```
latitude (°); longitude (°); altitude (m); tempo (s)
```

Crie uma função do tipo **GPS** que recebe uma linha lida do arquivo e retorne os valores numéricos dos dados decompostos em campos.

```
typedef struct{  
    float lat, lon, alt;  
    int tempo;  
} GPS;
```

O arquivo gerado pelo GPS e que será lido pelo Excel considera a vírgula como separador decimal, entretanto, ao programar em C, utilizamos o caractere ponto.

Assim, antes de separar os dados da linha lida do arquivo, ou após criar a nova linha que será gravada no arquivo de saída, devemos fazer a substituição dos caracteres.

```
void TrocarChar(char linha[],  
                char antigo,  
                char novo);
```

Após trocar o caractere de separação decimal, podemos criar uma função que separa as informações de cada linha, retornando-as no parâmetro valores

```
void SepararDados(char linha[], int quantidade,  
                  char separador, GPS *valores)
```

Na chamada dessa função, a quantidade deve ser 4 e o separador ‘.’.

Note que a função SepararInteiros pode ser alterada para resolver esse problema.

Cálculo da distância, em metros, entre dois pontos descritos em coordenadas geográficas (do tipo **GPS**):

$$dh = R_{\oplus} \cdot \text{acos} \left( \begin{array}{l} \text{sen}(\text{Lat}_A) \cdot \text{sen}(\text{Lat}_B) + \\ \text{cos}(\text{Lat}_A) \cdot \text{cos}(\text{Lat}_B) \cdot \text{cos}(\text{Lon}_A - \text{Lon}_B) \end{array} \right)$$

$$dv = \text{Alt}_B - \text{Alt}_A$$

$$\text{Dist}(A,B) = \sqrt{dh^2 + dv^2}$$

- sendo o raio da Terra:  $R_{\oplus} = 6372795,477598\text{m}$ .
- Não se esqueça de converter os ângulos Lat e Lon para radianos.



### Cálculo da velocidade instantânea, em km/h:

$$\text{velocidade}[\text{km/h}] = 3,6 \cdot \frac{\text{Dist}(P_{\text{Atual}}, P_{\text{Anterior}})[\text{m}]}{\text{Tempo}_{\text{Atual}} - \text{Tempo}_{\text{Anterior}}[\text{s}]}$$

- Ao final do processamento de cada linha do arquivo (leitura, cálculos e gravação), atualizar os dados do ponto Anterior com os valores do ponto Atual.

```
latitude (°);longitude (°);altitude (m);tempo (s); vel (km/h)
-24,177337;-47,024178;98;0;0
-24,177353;-47,024119;97;1;21,600000
-24,177335;-47,023955;105;2;64,800003
-24,177401;-47,023985;118;5;18,000000
```

Atenção! Ao ler dos dados do arquivo de entrada faça o processamento dos dados somente se a localização for diferente do ponto anterior **OU** o instante de tempo for diferente do anterior.

Note que o arquivo de saída poderá ter menos dados do que o arquivo de entrada gerado pelo GPS.

Mesmo assim será necessário atualizar os dados do ponto Anterior com o ponto Atual.

## Referências

GILLEN, D; CHANG, E; JOHNSON, D. Productivity benefits and cost efficiencies from its applications to public transit: the evaluation of AVL. In: BEKIARIS, E. NAKANISHI, Y. J. (Org.). **Economic Impacts of Intelligent Transportation Systems: Innovations and Case Studies**. Amsterdam: Elsevier, 2004. p.549-570.

DOLAZZI, L. R. M.; SCALCO, R. Sistema para gerenciamento das tarefas de funcionários utilizando aplicativo em *smartphone*. São Caetano do Sul, 2014. Relatório (Anteprojeto de Pesquisa). Escola de Engenharia Mauá.