

O presente projeto envolve a criação de uma biblioteca de classes respetivos métodos e testes que permitam gerir a informação relativa a trajetórias GPS que representam, viagens e respectivos consumos de combustível, energia, velocidade e uso de energia auxiliar<sup>1</sup>.

Os dados foram recolhidos de Novembro de 2017 a Novembro de 2018. Sendo a frota composta por um total de 384 automóveis pessoais (264 veículos a gasolina ICE, 93 veículos híbridos HEV e 27 veículos híbridos plug-in e eléctricos PHEV/EV) em Ann Arbor, Michigan, EUA.

O conjunto de dados é composto pela descrição e parâmetros de cada um dos 384 veículos (Dados Estáticos contidos nos ficheiros VED\_Static\_Data\_ICE&HEV.xlsx e VED\_Static\_Data\_PHEV&EV.xlsx) e pelos registos semanais de condução com marca temporal para os 384 veículos (Dados Dinâmicos: VED\_mmddyy\_week.csv onde mmddyy é o primeiro dia da semana a que os dados respeitam).

#### Dados Estáticos:

- Ficheiros: VED\_Static\_Data\_ICE&HEV.xlsx e VED\_Static\_Data\_PHEV&EV.xlsx
- Parâmetros de 384 veículos (264 a gasolina, 93 HEVs, e 27 PHEV/EVs):
- Colunas: VehId, Vehicle Type, Vehicle Class, Engine Configuration & Displacement, Transmission, Drive Wheels, Generalized\_Weight[lb]

#### Dados Dinâmicos:

- Ficheiros de dados semanais: VED\_mmddyy\_week.csv (mmddyy + 7 dias)
- Dados das viagens (Trip) de determinado veículo (VehId)
- Colunas: DayNum, VehId, Trip, Timestamp(ms), Latitude[deg], Longitude[deg], Vehicle Speed[km/h], MAF[g/sec], Engine RPM[RPM], Absolute Load[%], Outside Air Temperature[DegC], Fuel Rate[L/hr], Air Conditioning Power[kW], Air Conditioning Power[Watts], Heater Power[Watts], HV Battery Current[A], HV Battery SOC[%], HV Battery Voltage[V], Short Term Fuel Trim Bank 1[%], Short Term Fuel Trim Bank 2[%], Long Term Fuel Trim Bank 1[%], Long Term Fuel Trim Bank 2[%]

Notas: Cada combinação de VehID, Trip é única. DayNum Representa o número de dias decorridos desde uma data de referência. (DayNum 1 = Nov, 1st, 2017, 00:00:00, DayNum 1.5 = Nov, 1st, 2017, 12:00:00)

A ligação entre as entidades dos dados estáticos (dados de veículos) e dados dinâmicos (viagens) é feita através do VehId.

---

<sup>1</sup> VED (Vehicle Energy Dataset): <https://github.com/gsoh/VED/tree/master>

Usando as classes **árvore binária de pesquisa (BST)** e **árvore binária de pesquisa balanceada (AVL)** apresentadas nas aulas, desenvolva da forma **mais eficiente possível** as seguintes funcionalidades:

1. Carregar em BST ou AVL a informação relativa aos dados estáticos dos veículos e dados dinâmicos de viagens, permitindo efetuar uma pesquisa por VehId para obter os dados estáticos do veículo bem como todos os detalhes das viagens realizadas. Sem duplicar o carregamento de dados, a estrutura também deverá permitir obter para uma determinada viagem (Trip) todos detalhes e leituras da viagem e os detalhes do veículo associado.
2. Para um intervalo temporal entre dois DayNum, obter o máximo, mínimo e média dos parâmetros Vehicle Speed[km/h], Absolute Load[%], OAT[DegC] agregados por tipo de veículo (Vehicle Type).
3. Para um conjunto de viagens (Trip) ou intervalo de viagens obter a Latitude[deg] e Longitude[deg] de origem e destino de cada uma delas.
4. Para um conjunto de viagens identificadas pelo seu VehId obter para cada uma delas a viagem (Trip) com maior distância em linha reta entre origem e destino.
5. Com recurso à **2d-tree** e dadas duas coordenadas GPS de origem e destino, devolva os detalhes da viagem (Trip) cuja origem e destino são as mais próximas das coordenadas fornecidas.
6. Com recurso à 2d-tree devolva as Top-N viagens mais longas dentro de uma área geográfica dada por duas coordenadas GPS.

### **Normas**

- A avaliação do trabalho será feita principalmente em função das classes propostas, nomeadamente em termos da sua conformidade com o Paradigma da Programação por Objetos e **eficiência** das estruturas de dados usadas e funcionalidades solicitadas.
- O trabalho deverá ser realizado em **grupos de quatro alunos**.
- O projeto tem de ser desenvolvido em Java e todas as funcionalidades testadas através de testes unitários e usando os ficheiros de teste disponibilizados.
- É obrigatório o uso da ferramenta de **controle de versões Git**.
- O relatório deverá servir de ferramenta de avaliação posterior à apresentação. Nele devem apresentar o digrama de classes, **análise de complexidade** dos algoritmos das funcionalidades implementadas.
- O trabalho deve ser submetido no Moodle até às **24 horas do dia 5 de novembro**. A partir desta data a nota do trabalho será penalizada **10% por cada dia de atraso** e não se aceitam trabalhos **após dois dias** da data indicada.