

SEMANAS:

INÍCIO/APRESENTAÇÃO  
INTRODUÇÃO AOS ALGORITMOS E ESTRUTURAS DE DADOS  
ESTRUTURAS DE CONTROLO DE FLUXO EM C#  
CONCEITO DE CLASSE E OBJETO  
SEPARAÇÃO ENTRE INTERFACE PÚBLICA E IMPLEMENTAÇÃO PRIVADA  
ESTRUTURAS DE DADOS – INTRODUÇÃO  
ANÁLISE DE ALGORITMOS E COMPLEXIDADE  
ALGORITMOS DE ORDENAÇÃO  
ALGORITMOS DE ORDENAÇÃO (PARTE 2)  
PROVA DE AVALIAÇÃO  
LISTAS E FILAS  
TABELAS DE HASH  
TRATAMENTO DE COLISÕES  
ÁRVORES BINÁRIAS  
TIPOS GENÉRICOS E TRATAMENTO DE EXCEPÇÕES  
PROVA DE AVALIAÇÃO  
GRAFOS

# Algoritmos e Estruturas de Dados

Informática 2022/23



Docente: Ricardo Henriques .

Sumário:

- Enunciado do 2º trabalho de grupo

## Enunciado do 2º trabalho de grupo

Leia atentamente e em caso de dúvidas coloque por escrito, ao docente: [mhenriques@umaia.pt](mailto:mhenriques@umaia.pt) . Utilize o email institucional.

### Requisitos tecnológicos (obrigatórios)

- Desenvolva a parte prática do seu trabalho com base na constituição de uma solução *.NET Console Application*, em C#.
- Deve apenas recorrer às estruturas de dados criadas por si, *arrays* (vetores), listas ligadas, e tabelas de *hash*, e no contexto das alíneas e dos requisitos abaixo indicados (lógica de negócio).

### Lógica de negócio (requisitos mínimos para 15 valores)

Contexto: suponha que pediram ao seu grupo para implementar a funcionalidade do registo de veículos automóveis.

- O proprietário de um veículo é caracterizado, no mínimo, por:
  - Número de identificação fiscal portuguesa (campo único e obrigatório);
  - Nome completo (campo obrigatório);
  - Número de contacto (campo opcional);
  - Data de nascimento (campo obrigatório).
- O veículo é caracterizado, no mínimo, por:
  - Matrícula (campo único e obrigatório);
  - Marca (campo obrigatório);
  - Ano de construção (campo obrigatório).

O seu sistema deve implementar:

- i. O repositório dos proprietários numa **lista ligada simples**, ordenada pelo nome do proprietário.
- ii. O repositório dos veículos numa **tabela de hash com listas de colisão**, cujo tamanho da tabela (*array* de base), deve possuir 10 posições (ie. 10 listas de colisões).
- iii. A operação de compra de um veículo novo.



- iv. A operação de venda/compra de um veículo em 2ª mão.
- v. Ambos os repositórios devem permitir:
  - Inserção de um novo elemento;
  - Procura de um elemento (dada a chave) e devolução do valor (objeto);
  - Edição dos atributos de um elemento, com base na sua chave;
  - Remoção de um elemento.

### **Lógica de negócio (requisitos de valorização)**

- i. O repositório dos proprietários deve permitir implementar também a procura, por texto, sobre o nome (ou parte dele); deve devolver um *array* dos objetos (proprietários) que contém essa parte no seu nome.  
Nota: imprimir não é devolver. Devolver pressupõe um `return`.
- ii. A operação para registar o falecimento do proprietário.
- iii. Validações para coerência do sistema.

### **Entrega e prazos**

O grupo deve ser composto por um mínimo de 3 elementos e um máximo de 5.

#### **Elementos a entregar**

- 1. Um relatório de síntese, em formato *.pdf*, com:
  - Esboço ou interpretação do problema.
  - Diagrama de classes (exportado pelo Visual Studio 2022).
  - As assinaturas dos métodos da classe *Sistema* (ou equivalente).
- 2. Um ficheiro *.zip* com toda a solução C#.

#### **Datas de entrega/defesa**

- i. Submissão na plataforma *moodle* até 23h59 do dia 12 de Junho de 2023.
- ii. Apresentação e defesa no dia 13 de Junho de 2023.