



Projeto 2, Programação I

Projeto 2

```
-----  
| (1) Criar ficheiro navios_por_categoria.csv  
| (2) Mostrar relatorio das embarcacoes por categoria  
| (3) Inserir novas embarcacoes de passageiros  
| (4) Sair  
|-----  
| Opcao --> |
```

Trabalho realizado por:

- . Daniel Libório, nº14070, LEIC
- . Gonalo Miguel Rainho Taborda, nº14065, LEIC
- . Henrique Lima Domingos, nº14064, LEIC
- . Manuel Correia da Silva Roque, nº14061, LEIC

Professora:

- . Val3ria Magalh3es Pequeno

Datas:

- . Relat3rio -> terminado a
- . Pseudoc3digo -> 30/12/2023
- . C3digo -> terminado



2- Introdução:

- Neste relatório iremos falar sobre como foi o desenvolvimento do nosso projeto e o que nele foi abordado.
- Falando do pseudocódigo, elaboramo-lo inicialmente para estruturarmos o nosso código, para a lógica deste ser correta.
- O código, realizado na aplicação Visual Studio Code (VSC), em linguagem C, integra um menu apelativo ao utilizador, com 4 opções de escolhas, das quais 3 são realizar ações dadas pelo código através de funções e a última serve para o utilizador sair do programa.

3- Aplicação:

3.1)

➔ **Descrição do problema:** Elaborar uma aplicação desktop usando a linguagem C para fazer a gestão de embarcações conforme descritas no cenário a seguir.

➔ **Objetivo:** Consolidação do aprendizado do paradigma procedural e uso de ficheiros na linguagem C.

A nossa aplicação parte de um ficheiro dado pela professora, com o objetivo de organizar toda essa informação. Para coletarmos toda essa informação, utilizamos um sistema com tokens. Para isso utilizamos a função “ strtok “, pertencente à biblioteca <string.h>.

Para utilizarmos esta função, fizemos uma estrutura de repetição (while), até a string ser nula. Depois a função “ strtok “ é responsável por dividir uma string em várias substrings, com um delimitador. Neste caso, o delimitador utilizado foi “ ; “, porque é assim que funciona o ficheiro “ .csv “. Para separarmos a informação em categorias, utilizamos o método de comparação com estruturas condicionais “ ifs “.

O funcionamento do case 1 e do 2, consiste nestes dois pontos. Para o case 1, tivemos de criar ficheiros “ .csv “, que já se encontrava separada por categorias. Para o case 2, foi inserir, estes dados no terminal.

Outra funcionalidade da nossa aplicação é adicionar navios do tipo passageiros. Para isso, pedimos ao utilizador, o nome do navio, a bandeira, o indicativo de chamada, o comprimento, o calado máximo, o porto de registo, o armador, a boca de sinal, a velocidade máxima e a data de construção, pois já sabíamos de antemão que o tipo de navio é passageiro.

Por fim, o nosso programa apresenta a opção de saída do mesmo.

3.2)

Neste projeto, desenvolvemos um menu apelativo e simples para a compreensão fácil, fase aos conhecimentos dos utilizadores.

O nosso **menu** é composto, por 4 escolhas sendo estas as seguintes:

- 1: Criar ficheiro navios_por_categoria.csv
- 2: Mostrar relatório das embarcações por categoria
- 3: Inserir novas embarcações de passageiros
- 4: Sair

Explicação das escolhas do utilizador:

1) Criação de um ficheiro CSV (valores separados por pontos e vírgulas) é um ficheiro de texto com um formato específico que permite que os dados sejam guardados num formato estruturado de tabela. **Exemplo: EXCEL**

2) Organiza-se essa informação toda, agrupando as embarcações por categoria, a saber:

- Rebocador
- Passageiro
- Cargueiro
- Tanque
- Outros (qualquer navio que não se enquadrem nas categorias anteriores).

Das informações contidas no ficheiro, o navio agrupado em categorias deve-se buscar os seguintes dados:

- 1- Nome do navio,
- 2- bandeira,
- 3- tipo do navio,
- 4- indicativo de chamada,

- 5- comprimento,
- 6- calado máximo,
- 7- porto de registo,
- 8- armador,
- 9- boca sinal,
- 10- velocidade máxima (se esta for diferente de zero),
- 11- data da construção.

Os dados obtidos devem ser mostrados em um relatório, em que, dado a categoria da embarcação mostra as respetivas embarcações.

3) Esta opção permite inserir, na categoria de passageiros, novas embarcações, definidas pelo utilizador.

4) Esta opção permite ao utilizador fechar o programa.

Pormenor do código:

→ Usámos a biblioteca *"locale.h"* para definirmos a função:
setlocale(LC_ALL, "Portuguese");

Desta forma conseguimos utilizar acentos e "ç", de acordo com a linguagem portuguesa e considerámos isto essencial, porque existem um tipo de navio denominado "Batelão".

Executável:

→ gcc -Wall -Werror -pedantic -ansi -o Projeto2.exe Projeto2.c
→ ./Projeto2.exe

4- Conclusão

-> A nosso ver não tivemos tanto tempo para organizar as ideias quanto o primeiro projeto, por isso, tínhamos o tempo mais condicionado.

-> Este projeto era mais complexo, portanto exigia mais esforço e mais horas de trabalho dadas ao mesmo. No entanto no final, penso que o conseguimos realizar conforme nos foi pedido, apenas pensamos que o case 1 podia estar melhor, portanto:

-> O nosso balanço relativamente a este projeto é positivo pois conseguimos gerir bem o tempo que tínhamos para realizar as tarefas que nos foram propostas, com o máximo de eficácia e de concentração possível, menos na parte do case 1, como referido em cima. No entanto, consideramos que o projeto em si foi duro, pois exigiu-nos muito esforço.

-> Abordando de uma forma mais técnica, este projeto ajudou-nos imenso a entender a lógica de funções, vetores, structs, strings, ponteiros e sobretudo a de ficheiros, pois ao longo da elaboração do mesmo, com as pesquisas realizadas, ficamos a entender melhor a linguagem "C".