

Projeto Programação I – Fase 1

ESTG – Instituto Politécnico de Leiria

DEI – Departamento de Engenharia Informática

Licenciatura em Engenharia Informática

Daniel Brás – 2221453; José Branco - 2221424

## Objetivo

No âmbito da cadeira de Programação I, foi nos pedido para realizar um programa em linguagem C que auxilie na gestão da requisição de computadores portáteis pelos membros da comunidade académica do Politécnico.

Neste relatório temos como objetivo descrever e representar graficamente as estruturas de dados utilizadas no nosso projeto, assim como identificar algumas funcionalidades parcialmente ou incorretamente desenvolvidas e demonstrar e indicar a justificação de algumas das nossas opções de implementação tomadas.

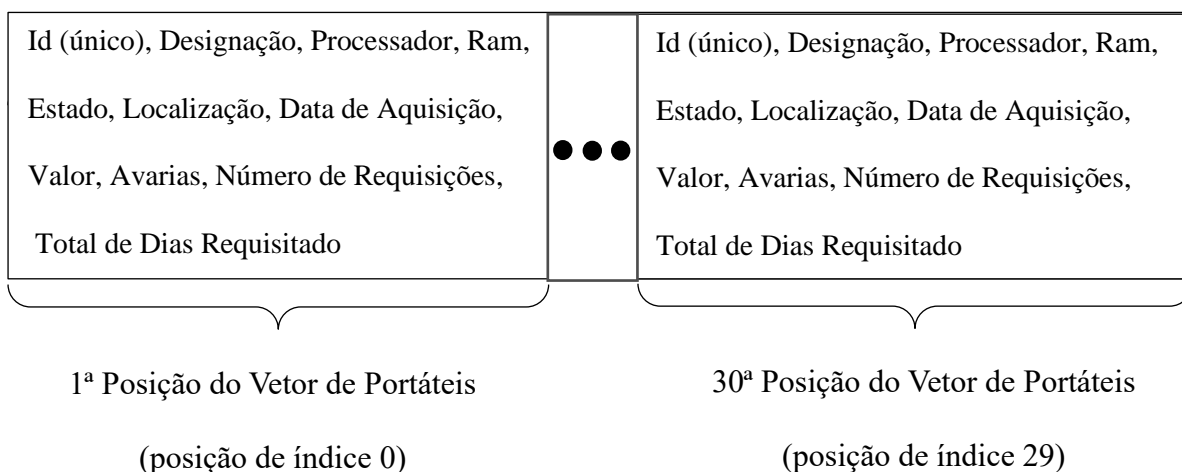
## Representação Gráfica e Descrição de Estruturas de Dados

Vetor Estático para Portáteis

Vetor Dinâmico para Requisições

Vetor Dinâmico para Avarias

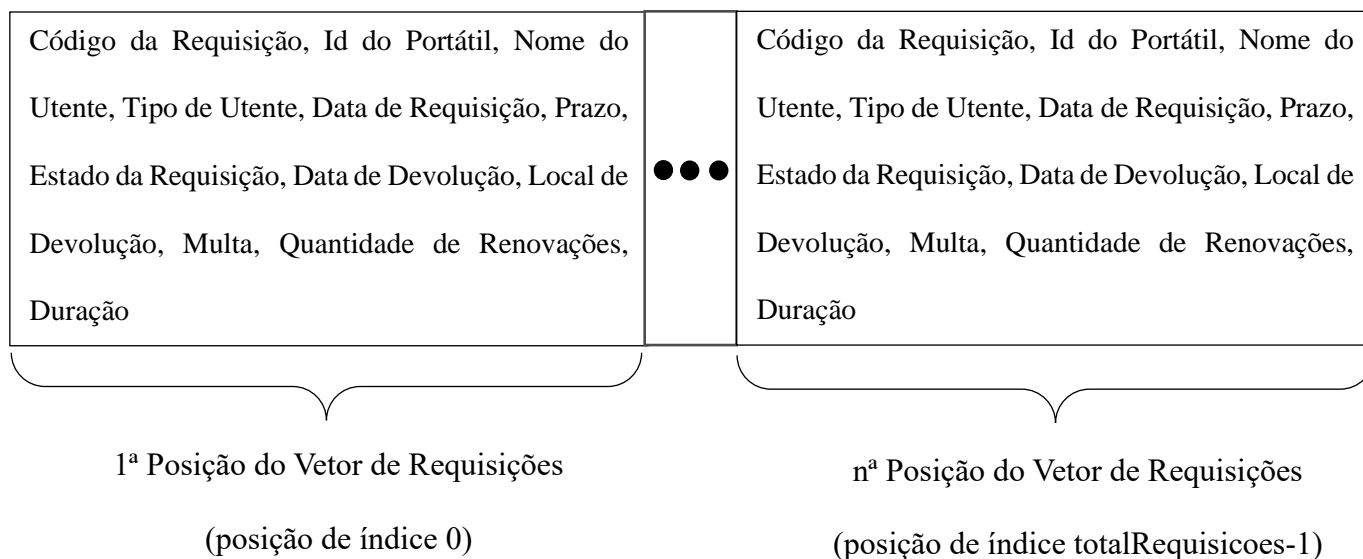
### Vetor Estático para Portáteis (Máx. 30 Portáteis)



Descrição: Para os portáteis, decidimos criar um vetor estático de estruturas, em que cada estrutura possui: o número de identificação do portátil (Id único: int de 1 a 30), a sua designação (string), processador (para i3,i5 ou i7 utilizámos os inteiros 3,5 e 7 respetivamente), RAM (GB: inteiros entre 1 e 100), estado (disponível, requisitado ou avariado: string), localização (Residências, Campus1, Campus2 ou Campus5: string), data de aquisição (tipoData do tipo: int/int/int), valor do equipamento (em euros: float), quantidade de avarias (int), quantidade de vezes que o mesmo foi requisitado (int) e a duração total de todas as suas requisições (em dias: int)

## Representação Gráfica e Descrição de Estruturas de Dados

### Vetor Dinâmico para Requisições (n Requisições)

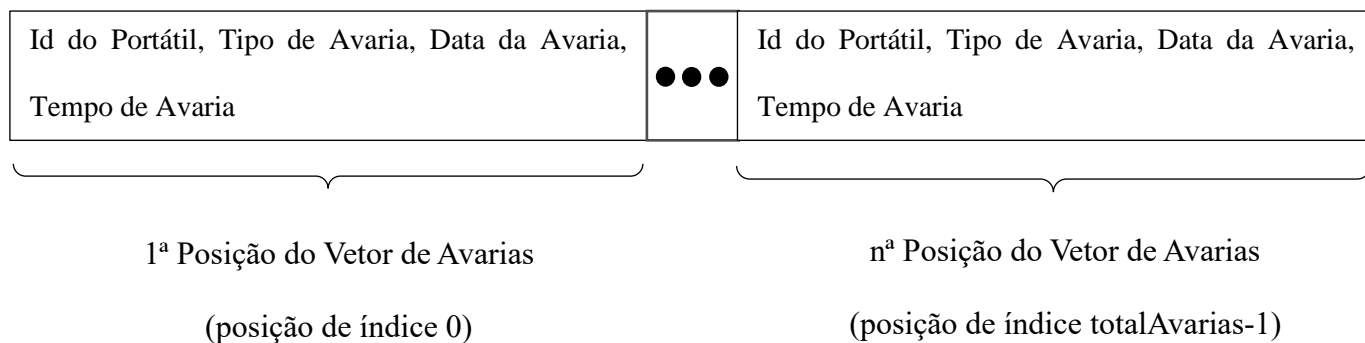


Descrição: Para as requisições, decidimos criar um vetor dinâmico de estruturas, em que cada estrutura possui: código da requisição (string), o número de identificação do portátil correspondente à requisição, nome do utente (string), tipo de utente (estudante, docente ou técnico administrativo: string), data de requisição (tipoData), prazo (tipoData), estado da requisição (ativa ou concluída: string), data de devolução (tipoData), local de devolução (Residências, Campus1, Campus2 ou Campus5: string), multa (em euros: int)\*, quantidade de renovações (int) e duração da requisição (em dias: int).

\*Nota: Embora a multa seja uma quantia em euros, decidimos colocar em formato de inteiro pois nunca será necessário utilizar casas decimais.

## Representação Gráfica e Descrição de Estruturas de Dados

### Vetor Dinâmico para Avarias



Descrição: Para as avarias, decidimos criar um vetor dinâmico de estruturas, em que cada estrutura possui: número de identificação do portátil correspondente à avaria, tipo de avaria (temporária ou permanente: string), data da avaria (tipoData), tempo de avaria (em dias: int).

## Justificação de Algumas Implementações Tomadas

- Para conseguirmos saber os dias de atraso numa devolução de um portátil, achámos que a melhor opção para calcular a diferença de dias entre duas datas seria criar uma função auxiliar, e assim, caso houvesse uma multa a pagar, seria mais fácil calcular o valor da mesma. Deste modo, encontrámos um código na internet no qual nos baseámos.<sup>1</sup>  
Já para somar 7 ou 30 dias para calcular o prazo de entrega de uma requisição, dada qualquer data, decidimos criar nós mesmos as duas funções utilizadas: “calculaData7” e “calculaData30”.
- Ao renovar ou registar avaria de um portátil requisitado, decidimos não chamar a função “devolver()” para não acontecer o caso do utilizador digitar outro código de portátil ou de requisição que não o do portátil que estava a renovar ou a registar a avaria. A desvantagem da nossa escolha é a repetição de algum código.

## Funcionalidades não Desenvolvidas

- A maior desvantagem do nosso programa é que, caso o utilizador escolha qualquer opção do menu, não pode voltar atrás com a escolha enquanto não preencher tudo o que lhe for pedido (a não ser que decida clicar no X do canto superior direito para fechar o programa).
- Outra desvantagem que o nosso programa apresenta é o caso do utilizador apenas clicar na tecla “ENTER” quando é pedido algum dado ou opção. Este não mostra nenhuma mensagem de erro mas stambém não faz nada, apenas dá o parágrafo (com excessão da função lerString que dá uma mensagem de erro).

## Referências

<sup>1</sup>Link: [https://www.youtube.com/watch?v=\\_-zam68lybE](https://www.youtube.com/watch?v=_-zam68lybE)