



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Princípios de Programação Procedimental

Engenharia Informática • 2024

Manual do Programador • Hospital

Trabalho realizado por:

Eduardo Marques - 2022231584

Sérgio Marques - 2022222096

Índice

Descrição do Projeto	2
Estrutura do Projeto	2
Funcionalidades da Aplicação	3
Conclusão	4

Descrição do Projeto

O projeto realizado consistiu no desenvolvimento de uma aplicação em C que faz a gestão dos doentes de um médico de um determinado hospital. Na construção desta aplicação, foram implementadas todas as funcionalidades pedidas, utilizando as estruturas de dados devidas e mecanismos de proteção adequados. Para armazenar os dados relevantes ao funcionamento do programa, durante a sua execução, foram utilizadas duas estruturas de dados: uma lista ligada que armazena todos os doentes e, dentro dessa lista, uma outra lista ligada para guardar todos os registos desse mesmo doente. Fora da aplicação, o armazenamento dos dados dos doentes do hospital foi efetuado utilizando ficheiros de texto.

Estrutura do Projeto

Como referido anteriormente, na realização deste projeto foram utilizadas duas listas ligadas que guardam os dados de cada doente e dos respetivos registos. Dentro destas listas, foi armazenada toda a informação pessoal dos doentes (ID, nome, data de nascimento, n.º cartão de cidadão, n.º telefone e email), bem como as informações dos seus registos (ID, data do registo, tensão máxima, tensão mínima, altura e peso). Assim que o programa é inicializado, todos estes dados são carregados dos ficheiros de texto para as respetivas listas.

A criação e manipulação destas listas foi elaborada com base nas funções desenvolvidas nas aulas e disponibilizadas pelos docentes, como a função de inserção ou a função de pesquisa que desempenharam um papel fundamental na inserção de dados nos nós das listas, sendo essenciais para um bom funcionamento do programa. Com o intuito de manter a integridade da aplicação, os nós das listas foram, previamente, alocados dinamicamente em memória. A alocação de memória foi, portanto, imprescindível para a realização eficaz de todas as operações constituintes do programa.

Na construção da aplicação foi, também, tida em conta a proteção de todos os dados que são armazenados nas listas quando efetuadas certas operações. Isto é, todos os dados introduzidos pelo utilizador na consola são sujeitos a uma verificação, que indica se estes dados são válidos para serem inseridos nas listas. Para isso, foram implementados algoritmos de verificação, com recurso a funções de manipulação de strings presentes em bibliotecas como a “*string.h*”.

De modo a evitar perdas da informação dos doentes, aquando do encerramento da aplicação, todos os dados são guardados de forma correta nos respetivos ficheiros de texto.

Funcionalidades da Aplicação

No menu da aplicação, o utilizador dispõe de várias funcionalidades que facilitam a gestão dos doentes:

- **Adicionar um novo doente**

Ao selecionar a primeira opção do menu, o utilizador tem a possibilidade de adicionar um novo doente introduzindo os seus dados. A aplicação garante que os mesmos são atribuídos a uma estrutura do tipo Doente e que são guardados na lista de doentes na posição correta através da função *ler_adicionar_doentes()*.

- **Eliminar um doente**

A eliminação de um doente é feita através do ID introduzido pelo utilizador. O programa está preparado para percorrer a lista dos doentes e eliminar o doente correto sem causar transtornos à integridade da lista, libertando a sua memória. Para isso, é chamada a função *ler_apagar_doentes()*.

- **Listar Doentes**

A terceira opção dá ao utilizador a possibilidade de listar todos os doentes disponíveis e as suas respetivas informações. O código desenvolvido percorre os nós da lista e acede às informações dos doentes contidos nos mesmos, imprimindo-os na consola através da função *imprime_doentes()*.

- **Listar Doentes com Tensões Altas**

Esta opção tem como objetivo mostrar ao utilizador os doentes com registo que contenham tensões máximas acima de um valor que é previamente introduzido, sendo que o programa fica encarregue de, numa primeira instância, ordenar de forma decrescente os doentes pela sua maior tensão máxima e, posteriormente, ordenar também os seus registo da mesma forma. Assim, esta informação é mostrada ao utilizador com a função *ler_imprime_tensoes_altas()*.

- **Listar Informação de um Doente**

Selecionando esta opção, o utilizador pode visualizar as informações pessoais de um determinado cliente, bem como os dados dos seus registo. Para imprimir na consola as informações do doente, será percorrida a lista dos doentes até ser encontrado o doente cujo ID corresponde ao ID requisitado pelo utilizador. Depois de encontrado o doente pretendido, a sua informação é mostrada na consola com recurso à função *ler_imprime_registros()*.

- Efetuar um Registo

A sexta opção disponível permite ao utilizador fazer um registo das medições efetuadas a um doente num determinado dia. O utilizador insere os dados relativos ao registo e é percorrida a lista de doentes com o intuito de encontrar um doente com o ID do registo. Caso seja encontrado esse doente, será inserido um novo registo na sua lista de registos. Para realizar esta operação, é utilizada a função *ler_efetuar_registros()*.

- Sair

Esta opção permite ao utilizador encerrar a aplicação. Quando selecionada, todos os dados de cada doente existente na lista dos doentes serão armazenados no respetivo ficheiro de texto. De seguida é libertada a memória ocupada pela lista dos doentes e é encerrada a aplicação, sendo enviada uma mensagem de despedida para a consola do utilizador.

Conclusão

Para concluir, o trabalho prático desenvolvido permitiu a criação de um programa que visa auxiliar um médico na gestão dos seus doentes, armazenando os seus dados de forma dinâmica e organizada com recurso a listas ligadas e ficheiros de texto. A manipulação de strings e a alocação dinâmica de memória foram igualmente importantes, garantindo a eficiência do programa. Dito isto, a realização deste trabalho permitiu-nos aplicar os conceitos lecionados e perceber que o uso de listas ligadas é uma ótima prática na resolução de problemas deste género.