

**Relatório**

**Algoritmos**

**Bruno Mendes**

**Xavier Monteiro**

**Marta Martinho**

**Desenvolvimento Web e Multimédia**

Polo de Famalicão, janeiro, 2021

Índice

[1. Introdução 1](#_Toc527567531)

[1.1. Contextualização 1](#_Toc527567532)

[1.2. Motivação e Objetivos 2](#_Toc527567533)

[2. Desenvolvimento 3](#_Toc527567531)

[2.1. Criar structs 3](#_Toc527567532)

[2.2. Criar um funcionario 4](#_Toc527567533)

[2.3. Inserir um funcionario 5](#_Toc527567534)

[2.4. Consultas Medicas 6](#_Toc527567534)

[2.5. Media das idades e vencimentos por genero 7](#_Toc527567534)

[2.6. Listar medicos e respetivos vencimentos 8](#_Toc527567534)

[2.7. Numero de compromissos de um enfermeiro 9](#_Toc527567534)

[2.8. Compromissos de um medico ou enfermeiro 10](#_Toc527567534)

[3. Conclusão 11](#_Toc527567531)

# Introdução

## Contextualização

Este trabalho foi desenvolvido no âmbito da Unidade Curricular de Algoritmos e Estruturas de dados do curso de CTESP em Desenvolvimento Web e Multimédia do IPCA em Famalicão. A realização do mesmo ocorreu durante o mês de janeiro.

O tema para este trabalho foi uma clínica. O presente projeto consistiu no desenvolvimento de uma aplicação em c, para gerir o pessoal das clínicas.

O trabalho foi proposto pela Professora Marta Marinho com o intuito de consolidar os conhecimentos adquiridos em aula, saber criar uma aplicação em c, utilizar structs para cada funcionário da clínica, trabalhar com agendas e aplicar funções como por exemplo media e soma.

No processo de desenvolvimento do trabalho, recorreu-se a ferramentas, tais como Microsoft Visual Studio Code já a linguagem utilizada foi o c.

## Motivação e Objetivos

Este trabalho tem, para além do inerente interesse académico, um forte interesse prático.

Do ponto de vista académico, é considerável a utilidade associada à implementação de uma rede de clínicas deste género. Através da realização deste trabalho, teve-se a possibilidade de contactar com os principais problemas que um sistema de clínicas.

Os principais objetivos deste trabalho são:

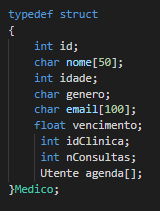
1. Apresentar um resumo que mostre a quantidade e média de idades dos funcionários por clínica, assim como a soma de todos os vencimentos, por género, de cada um dos grupos de funcionários;
2. Listar os médicos e os vencimentos agrupando-os por clínica, no final apresenta o somatório dos médicos listados;
3. Apresentar um resumo que apresente o número de compromissos de agenda de cada enfermeiro de uma clínica escolhida pelo utilizador;
4. Apresentar os compromissos de agenda do médico ou enfermeiro selecionado pelo utilizador, indicando qual o Utente associado a cada um dos compromissos.

# Desenvolvimento

## Criar structs para cada funcionário

Structs, também conhecidas como Registros, definem tipos de dados que agrupam variáveis sob um mesmo tipo de dado.

A ideia de usar uma struct é permitir que, ao armazenar os dados de uma mesma entidade, isto possa ser feito com uma única variável. Um exemplo de uma struct foi a de médicos que utilizamos no nosso trabalho.



Define o nome do novo tipo criado

Cria uma struct para armazenar os dados de um médico

Atributos do Medico

Após a criação do tipo, é possível declarar variáveis do tipo Medico, desta forma:

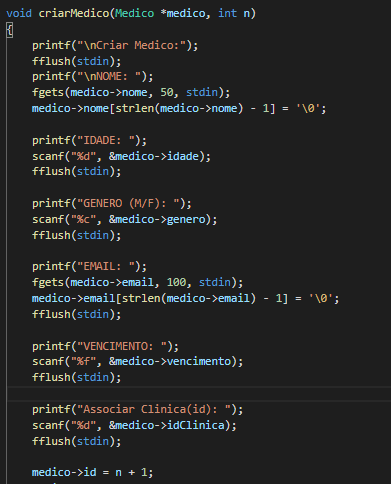
O TAMMEDICOS é uma variável constante usada para armazenar valores que não podem ser modificados durante a execução de um programa. Neste caso atribuímos um valor de 200 e depois criou se um vetor de tamanho TAMMEDICOS.



* 1. **Criar um funcionário**

A função utilizada para criar um medico é denominada de função com passagem por referência. Nesse tipo de função, os argumentos passam os endereços de memória para os parâmetros declarados na função. Sendo assim, os parâmetros que recebem os valores passados obrigatoriamente têm que ser ponteiros já que irão receber um endereço de memória. Nesta função fizemos com o utilizador inserisse os dados do medico e também pode associar a uma clínica através do id.

Depois de inserido o medico, o id do medico aumenta mais 1 automaticamente.

****

**Extras:**

A função *strlen()* retorna o número de caracteres contidos na *string* str, sem contar com o caractere delimitador \0.

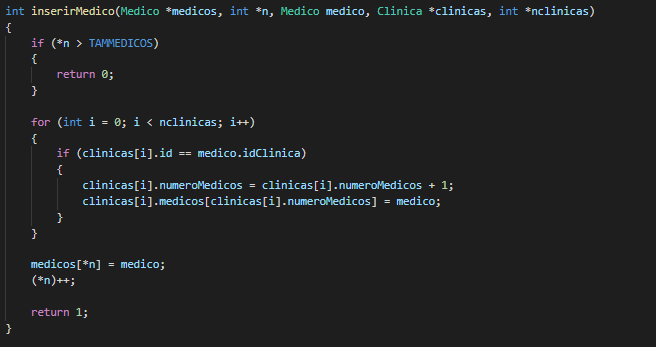
A função fflush(stdin) é usado para limpar o buffer do teclado.

Podemos usar para este problema também, o operador do scanf ou no fgets.

* 1. **Inserir um funcionário**

A função utilizada para criar um medico é denominada de função com return. Diferente do void que é uma função que não devolve, apenas executa já o int main() é uma função de devolve dependendo do tipo de variável retorna ou um numero inteiro ou um booleano.

Neste caso criamos uma função inteira em primeiro verifica se o numero de médicos é maior que o TAMMEDICOS (variável constante criada no topo do programa) caso seja return 0 que é igual FALSE, dando erro ao criar o medico, senão faz um ciclo for(vai percorrer cada coluna do vetor), depois vai verificar se o id da clinica é igual ao id do medico (ao criar o medico pede para inserir o numero de clinica), caso seja incrementa o numero de médicos. A seguir de estas validações forem todas verdadeiras vai return 1 que é igual a TRUE.

****

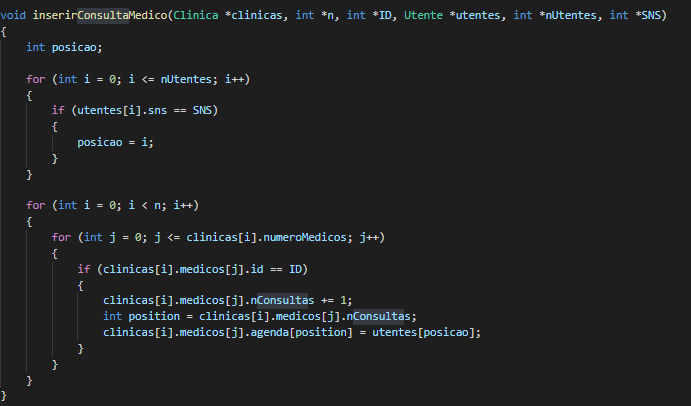
* 1. **Consultas Medicas**

Para agendar uma consulta medica, o utilizador pedia o sns do utente e id do medico para agendar uma consulta.

Nesta função, primeiro criamos uma variável posição, depois criamos um ciclo for para percorrer o array de utentes para a seguir verificar se existe um sns igual ao que o utilizador introduziu.

A seguir, criarmos um ciclo for para percorrer o array de médicos de uma clinica para a seguir verificar se existe um id igual ao que o utilizador introduziu, caso existe o numero de consultas incrementa mais um.

Depois criamos uma nova variável position atribuindo o valor do numero de consultas do medico de uma clinica.

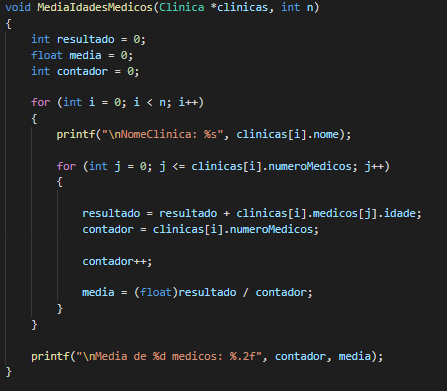


* 1. Media de idades e a soma dos vencimentos por género

Nesta função, começamos por criar variáveis para o resultado, media e contador atribuindo as, ás mesmas “0”.

Depois criamos um ciclo for para escrever o nome da clinica, a seguir criamos outro ciclo for para percorrer o array dos médicos de uma clinica, atribuindo assim o resultado ás idades dos médicos de uma determinada clinica. A seguir faz-se a media do resultado a dividir pelo contador.

A razão para criar este contador, foi para aparecer no printf da media, já que se puséssemos algum código com o i, iria dar erro.



* 1. Listar os médicos e os respetivos vencimentos

Nesta função criou se primeiro a variável resultado atribuindo “0” (mesmo raciocínio do ponto 2.5), depois fizemos um ciclo for para podermos fazer um género de uma tabela, listando assim o id, nome, idade, género, email e vencimento do medico.

A seguir, atribuímos o vencimento do medico ao resultado, para assim fazer a soma dos vencimentos dos médicos por clínica.

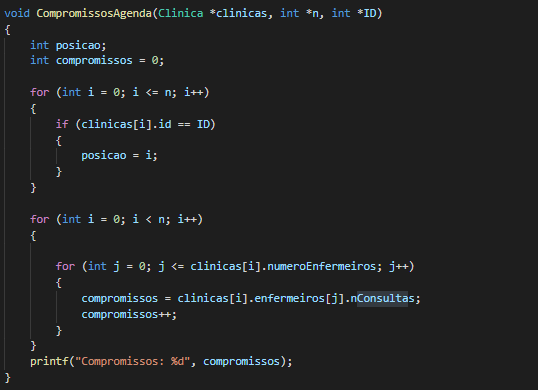


* 1. Número de compromissos de um enfermeiro

Nesta função criamos duas variáveis uma para contar os compromissos e outra a posição do array do id da clínica. Depois criamos um ciclo for para percorrer o array de clínicas para a seguir verificar se existe um id igual ao que o utilizador introduziu.

A seguir, criarmos um ciclo for para percorrer o array de enfermeiros de uma clínica para a seguir atribuir o número de consultas de um enfermeiro a compromissos, incrementa sempre mais um.

No final escreve o número de compromissos do enfermeiro que o utilizador introduziu.

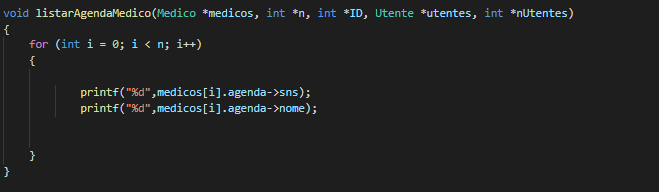


* 1. Compromissos de um medico

Nesta função primeiro pedimos ao utilizador se quer consultar um medico ou um enfermeiro (se inserisse o número 1 consultava o medico ou se inserisse o número 2 consultava o enfermeiro).

Depois pedia o id do medico, verificando a seguir se esse medico existia caso não existisse dava mensagem de erro e o utilizador tinha de introduzir um novo medico.

Na função de compromissos do medico mostrava os utentes que tinha na sua agenda.



# Conclusão

Com a elaboração deste trabalho, acabamos com bastante satisfação visto que superamos os objetivos propostos, superarmos algumas dificuldades em algumas partes do trabalho, mas tornou-se enriquecedor para os nossos conhecimentos. Para melhorias futuras, pretendemos fazer um registar e atualizar os utilizadores.