



INSTITUTO POLITÉCNICO DE VIANA DO CASTELO
ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO
ENGENHARIA INFORMÁTICA

Algoritmos e Estruturas de Dados

2020/2021

Gestão Hospitalar

Alunos:

João Amorim – N° 25355
Lucas Oliveira – N° 24753
Nelson Novais – N° 24844

Docentes:

Pedro Coutinho
Salvador Lima

1. Introdução

A atual pandemia de Covid-19 afetou gravemente as estruturas hospitalares, que viram um crescimento exponencial na afluência de doentes, sobrecarregando vários dos seus serviços hospitalares. Assim, era urgente encontrar uma solução a nível dos sistemas estruturais e funcionais. Para responder a essa necessidade foi realizado, no âmbito da Unidade Curricular de Algoritmos e Estruturas de Dados, uma aplicação com o intuito de facilitar a gestão dos médicos de um hospital, independentemente de estes estarem ou não diretamente relacionados com o tratamento de doentes covid.

Como a aplicação está relacionada com problemas da área da saúde era fundamental incluir nestas funcionalidades como, por exemplo: a possibilidade de fazer a distinção entre médicos especializados em Covid-19 e médicos não especializados; e ter a capacidade para fazer uma espécie de uma ficha (ficheiro de texto) para cada médico e para cada paciente, de modo a que o administrador possa ver as fichas dos profissionais de saúde em causa, e que estes possam ter acesso às fichas dos seus utentes. Para além disso, a fácil utilização da aplicação foi um dos principais objetivos, para que todos os utentes, mais velhos ou mais jovens, também conseguissem manusear a mesma para, por exemplo, ver os relatórios com a sua informação.

2. Desenvolvimento do projeto

Primeiramente, foi introduzido no DEV C++ o código para a criação dos menus do administrador, dos médicos e dos pacientes, onde estão disponíveis as diferentes opções que permitem aos utilizadores fazerem uso do programa e, quando acabarem a sua utilização, encerrá-lo.

Aquando da criação dos menus, tendo sido este o primeiro passo para a elaboração deste projeto, começamos a trabalhar nos principais objetivos para o funcionamento da aplicação em estudo. O menu do administrador permitir-nos-á inserir, ou remover médicos, e fazer uma listagem dos mesmos, para que haja um controlo sobre o número de trabalhadores no hospital. Através deste menu, também é possível remover utentes já registados, justificações, receitas ou até mesmo relatórios.

O menu que será utilizado pelos profissionais de saúde terá a função de inserir e listar novos pacientes, com o objetivo de, no fim, haver uma lista completa de todos os doentes que se encontram a ser tratados. Emitir receitas, internamentos, e informação sobre futuras consultas e medicação a realizar pelos utentes, também serão funcionalidades deste menu. Para além disso, permite realizar justificações de presença para o caso de um doente faltar ao trabalho para poder comparecer a uma consulta, e, também, relatórios clínicos com o intuito de terem na sua posse informação sobre determinado paciente. Acessos a informações sobre outras consultas também serão facultados.

Por último, o menu dos utentes deixa-os consultar os seus relatórios.

Deste modo, o administrador do programa começa por fazer as fichas de cada médico, fichas essas que ficam guardadas em ficheiros de texto para poderem ser acedidas posteriormente. A partir do menu dos médicos é possível inserir na aplicação novos pacientes, através da submissão da informação pessoal dos mesmos. Estas fichas também serão guardadas em ficheiros de texto, dando a possibilidade de serem acedidas pelos profissionais de saúde quando necessário. No que toca aos utentes, estes também podem utilizar a aplicação para poderem, por exemplo, ver os seus relatórios.

A seguir segue parte do código que utilizámos para realizar a aplicação pretendida.

```
printf("\n");
do{
    printf("\n\n");
    printf("\t|*****|\n"
"\t|****      Menu      ****|\n"
"\t|*****|\n"
"\t|****      1 - ADMIN      ****|\n"
"\t|****      2 - MÉDICO      ****|\n"
"\t|****      3 - UTENTE      ****|\n"
"\t|****      0 - SAIR      ****|\n"
"\t|*****|\n");
}
```

Ilustração 1 - Menu Geral

```
printf("\nSelecione a opção: ");
scanf("%d", &opcao);

switch(opcao){
    case 1:{
        admin(user, ficheiro, a, k, pass, g);
        menuAdmin(opcaoAdmin, opcaoCovid, c, ficheiro, w, x);

        break;
    }
}
```

Ilustração 2 - Acesso ao Menu do Administrador

```
case 2:{
    //medico(opcaoMedico, idMedico);
    iSessaoMedico(userm, passm, str1, str2, k, g, a, ficheiro);
    menuMedico(opcaoMedico, m, ficheiro, doenteCovid, w, z, q, nomeUtenteListar, a, k, y, s, h);
    break;
}
```

Ilustração 3 - Acesso ao Menu do Médico

```
case 3:{
    iSessaoUtente(userm, passm, str1, str2, k, g, a, ficheiro, opcaoUtente, v);
    //ConsultarRelatorio(opcaoUtente, userm, v, ficheiro);
    break;
}
}while(opcao != 0);
```

Ilustração 4 - Acesso ao Menu do Utente

```

void menuAdmin(int opcaoAdmin, char opcaoCovid, struct sessao c, char ficheiro[C], char w[100], char x[100]){
do{
    printf("\n\n");
    printf("\t *****\n"
           "\t ****          ADMIN          ****\n"
           "\t *****\n"
           "\t **** 1 - Inserir Médico  ****\n"
           "\t **** 2 - Listar Médicos  ****\n"
           "\t **** 3 - Remover        ****\n"
           "\t **** 0 - SAIR           ****\n"
           "\t *****\n");
}

```

Ilustração 5 - Menu do Administrador

```

switch(opcaoAdmin){
case 1:{
    FILE*fp;
    printf("\nÉ especializado em covid-19?(S)im/(N)ão: ");
    scanf("%s", &opcaoCovid);

    switch(opcaoCovid){
    case 'S':
    case 's':{
        sequencia("\nIntroduza o nome do médico: ", c.nome, 50);
        c.id = inteiro("\nIntroduza o nº interno do médico: ");
        sequencia("\nIntroduza a password do médico: ", c.pass, 50);
        sequencia("\nIntroduza a localidade do médico: ", c.localMedico, 100);
        c.nExperiencia = inteiro("\nIntroduza o nº de anos de Experiência: ");
        c.salario = inteiro("\nIntroduza o valor do salário: ");
        c.teleMedico = inteiro("\nIntroduza o contacto telefónico do médico: ");
        sprintf(ficheiro, ".\\Medico\\medico_%s.txt", c.nome);
        fp = fopen(ficheiro, "w");

        fprintf(fp, "\nNome: %s\nNº: %d\nPassword: %s\nEspecialidade: Covid-19\nLocalidade: %s"
                "\nExperiência: %d\nSalário: %d\nTelefone: %d", c.nome, c.id, c.pass, c.localMedico, c.nExperiencia, c.salario, c.teleMedico);
        fprintf(fp, "\n");
        fclose(fp);
        fp= fopen(".\\Medico\\MedicoCovid.txt", "a");
        fprintf(fp, "%s\n%d", c.nome, c.id);
        fclose(fp);
        break;
    }
}
}

```

Ilustração 6 - Criação de um Médico especializado em Covid-19

```

    case 'N':
    case 'n':{
        sequencia("\nIntroduza o nome do médico: ", c.nome, 50);
        c.id = inteiro("\nIntroduza o nº interno do médico: ");
        sequencia("\nIntroduza a especialidade do médico: ", c.especialidade, 50);
        sequencia("\nIntroduza a password do médico: ", c.pass, 50);
        sequencia("\nIntroduza a localidade do médico: ", c.localMedico, 100);
        c.nExperiencia = inteiro("\nIntroduza o nº de anos de Experiência: ");
        c.salario = inteiro("\nIntroduza o valor do salário: ");
        c.teleMedico = inteiro("\nIntroduza o contacto telefónico do médico: ");
        sprintf(ficheiro, ".\\Medico\\medico_%s.txt", c.nome);
        fp = fopen(ficheiro, "w");
        fprintf(fp, "\nNome: %s\nNº: %d\nPassword: %s\nEspecialidade: %s\nLocalidade: %s"
                "\nExperiência: %d\nSalário: %d\nTelefone: %d", c.nome, c.id, c.pass, c.especialidade, c.localMedico, c.nExperiencia, c.salario, c.teleMedico);
        fprintf(fp, "\n");
        fclose(fp);
        fp= fopen(".\\Medico\\MedicoNormal.txt", "a");
        fprintf(fp, "%s\n%d", c.nome, c.id);
        fclose(fp);
        break;
    }
    default:{
        erroOpcao(t);
        break;
    }
}
}
}

```

Ilustração 7 - Criação de um Médico não especializado em Covid-19

```

case 2:{
    FILE*fp;
    printf("\n");
    printf("\tMédicos Covid Registrados no sistema: ");
    fp = fopen(".\\Medico\\MedicoCovid.txt", "r");
    if(fp != NULL){
        fread(w, sizeof(w), 100, fp);
        printf("\n%s", w);
        printf("\n\n");
    }
    fclose(fp);

    printf("\tMédicos Normais Registrados no Sistema: ");
    fp = fopen(".\\Medico\\MedicoNormal.txt", "r");
    if(fp != NULL){
        fread(x, sizeof(x), 100, fp);
        printf("\n%s", x);
        printf("\n\n");
    }
    fclose(fp);
    break;
}

```

Ilustração 8 - Listagem dos Médicos

```

printf("\t|*****|\n"
       "\t|      ADMIN      |\n"
       "\t|*****|\n"
       "\t|      1 - Remover Médico      |\n"
       "\t|      2 - Remover Utente      |\n"
       "\t|      3 - Remover Justificação |\n"
       "\t|      4 - Remover Receita     |\n"
       "\t|      5 - Remover Relatório   |\n"
       "\t|      0 - SAIR                |\n"
       "\t|*****|\n");

```

Ilustração 9 - Menu de remoção do Administrador


```

c.remove = inteiro("\nOpção: ");
switch(c.remove){
    case 1:{
        sequencia("\nNome da Pessoa a remover: ", c.removerFicheiro, 100);
        sprintf(ficheiro, ".\\Medico\\medico_%s.txt", c.removerFicheiro);

        if (remove(ficheiro) == 0)
            printf("\nFicheiro Removido com sucesso");
        else
            printf("\nFicheiro Não Removido");

            break;
        }
    case 2:{
        sequencia("\nNome da Pessoa a remover: ", c.removerFicheiro, 100);
        sprintf(ficheiro, ".\\Utente\\Utente_%s.txt", c.removerFicheiro);

        if (remove(ficheiro) == 0)
            printf("\nFicheiro Removido com sucesso");
        else
            printf("\nFicheiro Não Removido");

            break;
        }
    case 3:{
        sequencia("\nNome da Pessoa a remover: ", c.removerFicheiro, 100);
        sprintf(ficheiro, ".\\Justificacao\\Utente_%s.txt", c.removerFicheiro);

        if (remove(ficheiro) == 0)
            printf("\nFicheiro Removido com sucesso");
        else
            printf("\nFicheiro Não Removido");

            break;
        }
}

```

Ilustração 10 - Remoção de Médicos, Utentes e Justificações

```

        case 4:{
            sequencia("\nNome da Pessoa a remover: ", c.removerFicheiro, 100);
            sprintf(ficheiro, ".\\Receitas\\Utente_%s.txt", c.removerFicheiro);

            if (remove(ficheiro) == 0)
                printf("\nFicheiro Removido com sucesso");
            else
                printf("\nFicheiro Não Removido");

                break;
            }
        case 5:{
            sequencia("\nNome da Pessoa a remover: ", c.removerFicheiro, 100);
            sprintf(ficheiro, ".\\Relatorio\\Utente_%s.txt", c.removerFicheiro);

            if (remove(ficheiro) == 0)
                printf("\nFicheiro Removido com sucesso");
            else
                printf("\nFicheiro Não Removido");

                break;
            }
        }
    }while(c.remove!=0);
}
}while(opcaoAdmin != 0);
}

```

Ilustração 11 - Remoção de Receitas e de Relatórios

```

void menuMedico(int opcaoMedico, struct utente m , char ficheiro[C], char doenteCovid, char w[100], char z[100],char q[100] ,
char nomeUtenteListar[100], char a[50], int k, char y[50], int s, char h[100]){
do{
printf("\n\n");
printf("\t*****\n"
\t**** Médico ****\n"
\t*****\n"
\t**** 1 - Inserir Paciente ****\n"
\t**** 2 - Listar Pacientes ****\n"
\t**** 3 - Emitir Receita ****\n"
\t**** 4 - Emitir Internamento ****\n"
\t**** 5 - Justificação ****\n"
\t**** 6 - Relatório Clínico ****\n"
\t**** 7 - Consultar informações de outras consultas ****\n"
\t**** 0 - Sair ****\n"
\t*****\n");

```

Ilustração 12 - Menu dos Médicos

```

opcaoMedico = inteiro("Escolha a sua opção: ");
switch(opcaoMedico){
case 1:{
FILE*fp;
printf("\nDoente Covid-19?(S)im ou (N)ão: ");
scanf("%s", &doenteCovid);
switch(doenteCovid){
case 'N':
case 'n':{
sequencia("\nNome do Paciente: ", m.nomeUtente, 100);
sequencia("\nPassword: ", m.password, 50);
sequencia("\nSexo: ", m.sexo, 50);
m.idade = inteiro("\nIdade: ");
sequencia("\nLocalidade: ", m.localUtente, 100);
m.teleUtente = inteiro("\nTelefone: ");
m.nPaciente = inteiro("\nNº de Utente: ");
sequencia("\nProblema de Saúde: ", m.ProblSaude, 100);
sequencia("\nSituação Clínica: ", m.SituacaoClinica, 100);
sequencia("\nData da Primeira Consulta: ", m.datada1Consulta, 20);
sequencia("\nData da Última Consulta: ", m.datadaUltimaConsulta, 20);
sequencia("\nData da Próxima Consulta: ", m.datadaProximaConsulta, 20);

sprintf(ficheiro, ".\\Utente\\Utente_%s.txt", m.nomeUtente);
fp = fopen(ficheiro, "w");

fprintf(fp, "\nNome: %s\nPassword: %s\nSexo: %s\nIdade: %d\nLocalidade: %s\nTelefone: %d"
"\nNº de Utente: %d\nProblema de Saúde: %s\nSituação Clínica: %s"
"\nData da 1ªConsulta: %s\nData da Última Consulta: %s"
"\nData da Próxima Consulta: %s"
, m.nomeUtente, m.password, m.sexo, m.idade, m.localUtente, m.teleUtente, m.nPaciente, m.ProblSaude, m.SituacaoClinica, m.datada1Consulta, m.datadaUltimaConsulta, m.datadaProximaConsulta);
fprintf(fp, "\n");
fclose(fp);
fp= fopen(".\\Utente\\UtentesNaoCovid.txt", "a");
fprintf(fp, "%s", m.nomeUtente);
fclose(fp);

break;
}
}
}

```

Ilustração 13 - Criação de doente sem Covid-19

```

case 'S':
case 's':{
sequencia("\nNome do Paciente: ", m.nomeUtente, 100);
sequencia("\nPassword: ", m.password, 50);
sequencia("\nSexo: ", m.sexo, 50);
m.idade = inteiro("\nIdade: ");
sequencia("\nLocalidade: ", m.localUtente, 100);
m.teleUtente = inteiro("\nTelefone: ");
m.nPaciente = inteiro("\nNº de Utente: ");
sequencia("\nSituação Clínica: ", m.SituacaoClinica, 100);
sequencia("\nData da Primeira Consulta: ", m.datada1Consulta, 20);
sequencia("\nData da Última Consulta: ", m.datadaUltimaConsulta, 20);
sequencia("\nData da Próxima Consulta: ", m.datadaProximaConsulta, 20);

sprintf(ficheiro, ".\\Utente\\Utente_%s.txt", m.nomeUtente);
fp = fopen(ficheiro, "w");

fprintf(fp, "\nNome: %s\nPassword: %s\nSexo: %s\nIdade: %d\nLocalidade: %s\nTelefone: %d"
"\nNº de Utente: %d\nProblema de Saúde: Covid-19\nSituação Clínica: %s"
"\nData da 1ªConsulta: %s\nData da Última Consulta: %s"
"\nData da Próxima Consulta: %s"
, m.nomeUtente, m.password, m.sexo, m.idade, m.localUtente, m.teleUtente, m.nPaciente, m.SituacaoClinica, m.datada1Consulta, m.datadaUltimaConsulta, m.datadaProximaConsulta);
fprintf(fp, "\n");
fclose(fp);
fp= fopen(".\\Utente\\UtentesCovid.txt", "a");
fprintf(fp, "%s", m.nomeUtente);
fclose(fp);

break;
}
}

```

Ilustração 14 - Criação de doente com Covid-19


```

case 2:{
    FILE*fp;
    printf("\n\tListagem de Utentes com Covid-19");
    fp=fopen("../Utente/UtentesCovid.txt","r");
    fread(w, sizeof(w),100,fp);
    printf("\n%s", w);
    fclose(fp);
    printf("\n\tListagem de Utentes Não Covid-19");
    fp=fopen("../Utente/UtentesNaoCovid.txt","r");
    fread(q, sizeof(q),100,fp);
    printf("\n%s", q);
    fclose(fp);
    sequencia("\nNome do Utente: ", nomeUtenteListar , 100);
    sprintf(ficheiro, "../Utente/Utente_%s.txt", nomeUtenteListar);
    fp = fopen(ficheiro, "r");
    if(fp != NULL){
        fread(z, sizeof(z),100,fp);
        printf("\n%s\n", z);
    }
    fclose(fp);
    printf("\n");
    break;
}

```

Ilustração 15 - Listagem de Utentes

```

case 3:{
    FILE*fp;
    printf("\nReceita");
    sequencia("\nNome do Utente: ", m.receitaUtente, 100);
    sprintf(ficheiro, "..\\Utente\\Utente_%s.txt", m.receitaUtente);

    if((fp = fopen(ficheiro, "r") )!= NULL){

        sequencia("\nNome(s) do(s) Medicamento(s): ", m.medicamento, 200);
        //fp = fopen(ficheiro, "w");

        sprintf(ficheiro, "..\\Receitas\\Utente_%s.txt", m.receitaUtente);
        fp = fopen(ficheiro, "w");
        fprintf(fp, "\nNome: %s\nNome(s) do(s) Medicamento(s): %s\nAssinatura do Médico: \n", m.receitaUtente, m.medicamento);
        fclose(fp);

        sprintf(ficheiro, "..\\Consultas\\Utente_%s.txt", m.receitaUtente);
        fp = fopen(ficheiro, "a+");
        fprintf(fp, "\nNome(s) do(s) Medicamento(s): %s", m.medicamento);
        fprintf(fp, "\n");
        fclose(fp);
    }

    else{
        printf("\nUtente não existente");
    }

    break;
}

```

Ilustração 16 - Emissão de Receitas

```

case 4:{
    FILE*fp;
    sequencia("\nNome do Utente: ", m.nomeInternamento, 100);
    sprintf(ficheiro, ".\\Utente\\Utente_%.txt", m.nomeInternamento);

    if((fp = fopen(ficheiro, "a") )!= NULL){
        fprintf(fp, "\nEste paciente está Internado");
        printf("\nEste paciente está Internado");
    }else{
        printf("\nUtente não existente");
    }
    fclose(fp);
    break;
}

```

Ilustração 17 - Emissão de Internamento

```

case 5:{
    FILE*fp;
    sequencia("\nNome do Utente: ", m.nomeJustificacao, 100);
    sprintf(ficheiro, ".\\Utente\\Utente_%.txt", m.nomeJustificacao);

    if((fp = fopen(ficheiro, "r") )!= NULL){

        sequencia("\nMotivo da Falta: ", m.motivoFalta, 300);
        sequencia("\nData da Falta: ", m.dataFalta, 20);
        fp = fopen(ficheiro, "w");

        sprintf(ficheiro, ".\\Justificacao\\Utente_%.txt", m.receitaUtente);
        fp = fopen(ficheiro, "w");
        fprintf(fp, "\nNome: %s\nMotivo da falta: %s\nData da Falta: %s\nAssinatura do Médico: \n", m.nomeJustificacao, m.motivoFalta, m.dataFalta);
        fprintf(fp, "\n");
        fclose(fp);
    }else{
        printf("\nUtente não existente");
    }
    break;
}

```

Ilustração 18 – Justificações

```

case 6:{
    FILE*fp;
    sequencia("\nNome do Utente: ", m.nomeUtente, 100);
    sprintf(ficheiro, ".\\Utente\\Utente_%.txt", m.nomeUtente);

    if((fp = fopen(ficheiro, "r") )!= NULL){
        sequencia("\nRelatório Clínico: ", m.relatorioClinico, 1000);

        sprintf(ficheiro, ".\\Relatorio\\Utente_%.txt", m.nomeUtente);
        fp = fopen(ficheiro, "w");
        fprintf(fp, "\nNome: %s\nRelatório Clínico: %s", m.nomeUtente, m.relatorioClinico);
        fprintf(fp, "\n");
        fclose(fp);

        sprintf(ficheiro, ".\\Consultas\\Utente_%.txt", m.nomeUtente);
        fp = fopen(ficheiro, "a+");
        fprintf(fp, "\nRelatório Clínico: %s", m.relatorioClinico);
        fprintf(fp, "\n");
        fclose(fp);
    }else{
        printf("\nUtente não existente");
    }
    break;
}

```

Ilustração 19 - Relatórios Clínicos

```

case 7:{
    sequencia("\nIntroduza o nome do Utente que pretende consultar: ", m.nomeUtente, 100);
    FILE*fp;
    sprintf(ficheiro, ".\\Consultas\\Utente_%s.txt", m.nomeUtente);
    fp=fopen(ficheiro, "r");
    if(fp == NULL){
        printf("erro");
    }else{

        fread(h, sizeof(h), 100, fp);
        printf("\n%s", h);
    }
    fclose(fp);

    break;
}

```

Ilustração 20 - Ver informação sobre outras consultas

```

printf("\n\n");
printf("\t|*****|\n"
        "\t|****          UTENTE          ****|\n"
        "\t|*****|\n"
        "\t|****  1 - Consultar Relatório  ****|\n"
        "\t|****  0 - SAIR                ****|\n"
        "\t|*****|\n");

opcaoUtente = inteiro("\nQual a sua opção: ");

    FILE*fp;
    sprintf(ficheiro, ".\\Relatorio\\Utente_%s.txt", userm);
    fp=fopen(ficheiro, "r");
    if(fp == NULL){
        printf("erro");
    }else{

        fread(v, sizeof(v), 100, fp);
        printf("\n%s", v);
    }
    fclose(fp);

```

Ilustração 21 - Menu do Utente

Structs Utilizados:

```
struct sessao{
    char nome[100];
    int id;
    char pass[50];
    char especialidade[50];
    int nExperiencia;
    int salario;
    int teleMedico;
    char localMedico[100];
    char removerFicheiro[100];
    int remove;
};
```

```
struct utente{
    char nomeUtente[100];
    int nPaciente;
    int idade;
    char sexo[50];
    char localUtente[100];
    char ProblSaude[100];
    char SituacaoClinica[100];
    char nomedoMedico[100];
    int teleUtente;
    char datada1Consulta[20];
    char dadtadaUltimaConsulta[20];
    char datadaProximaConsulta[20];
    char medicamento[200];
    char receitaUtente[100];
    char motivoFalta[300];
    char dataFalta[20];
    char password[50];
    char relatorioClinico[1000];
    char nomeInternamento[100];
    char nomeJustificacao[100];
};
```

3. Conclusão

Neste projeto, onde nos foi proposto desenvolver um programa que nos permitisse gerir os médicos e utentes de um hospital, fizemos uma aplicação que pudesse ser utilizada tanto pelos médicos como pelos próprios utentes.

Existe uma secção do programa dedicada para cada grupo, no entanto, é gerido por um certo administrador, que também tem a tarefa de criar a ficha de admissão dos profissionais de saúde. Também consegue remover informação da aplicação, como, por exemplo, utentes, justificações e relatórios clínicos.

Os médicos, que certamente serão o grupo que utilizará mais esta aplicação, conseguem inserir novos pacientes, podendo listá-los, para saberem que doentes é que estão no processo de tratamento, emitir receitas e internamentos, justificações, caso um paciente tenha de faltar, por exemplo, ao seu emprego para ter de ir ao hospital, e também podem ver relatórios clínicos. Conseguem até ver informação sobre outras consultas de pacientes. Os próprios utentes também podem utilizar a aplicação para ver, tendo como exemplo, os seus relatórios clínicos.