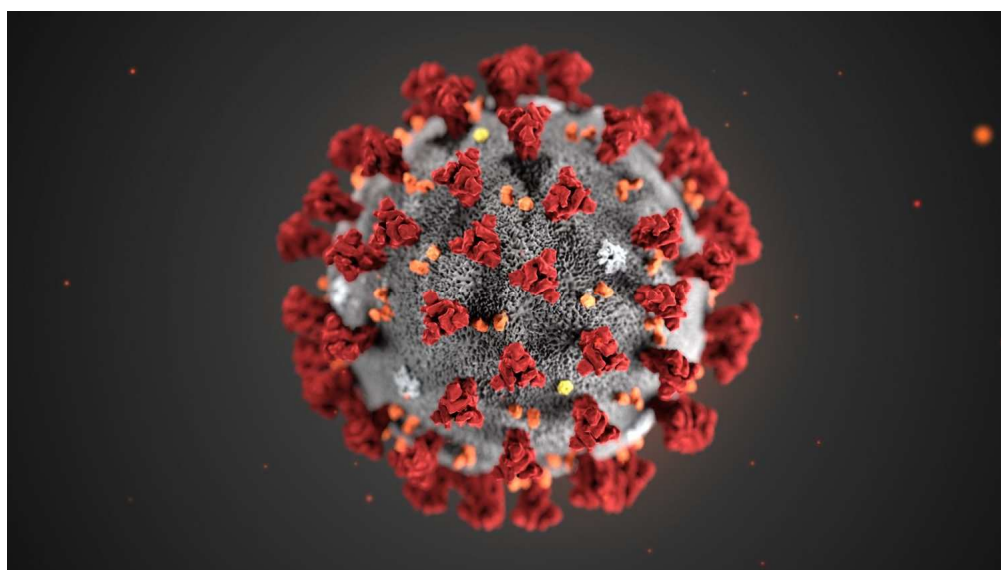


# SIMULAÇÃO DA PROPAGAÇÃO DE VÍRUS



15/06/2020

## Programação – Trabalho Prático

Implementação em *C standard* de um programa que simula, de forma simplificada, a propagação de um vírus entre uma população. Neste relatório serão apresentadas as principais estruturas de dados e as estruturas dinâmicas implementadas, com a respetiva justificação das escolhas feitas. Será também apresentado um pequeno manual de utilização.

## Conteúdo

<b>ESTRUTURAS DE DADOS .....</b>	<b>2</b>
<b>ESTRUTURAS DINÂMICAS .....</b>	<b>3</b>
<b>IMPLEMENTAÇÃO DO PROGRAMA .....</b>	<b>4</b>
FASE DE PREPARAÇÃO .....	4
FASE DE SIMULAÇÃO.....	5
<b>MANUAL DE UTILIZAÇÃO .....</b>	<b>10</b>
FASE DE PREPARAÇÃO .....	10
FASE DE SIMULAÇÃO.....	11

## PROGRAMAÇÃO – TRABALHO PRÁTICO

### ESTRUTURAS DE DADOS

```
typedef struct doentes pessoas, *pno;
struct doentes{
    char nome[MAX];
    int idade;
    char estado[1];
    int dias;
    pno prox;
};
```

A estrutura “pessoas” contém as características das pessoas, nomeadamente:

- nome, que é um identificador alfanumérico único;
- idade;
- estado, que pode ser **Saudável**, **Doente** ou **Imune**;
- dias, caso esteja Doente existe informação de há quantos dias tem o vírus;
- \*pno, este ponteiro permite apontar para uma pessoa.

```
typedef struct sala local, *plocal;
struct sala{
    int id;
    int capacidade;
    int liga[3]
};
```

A estrutura “local” foi utilizada apenas para ler dados de um ficheiro binário e passá-los para um vetor dinâmico de estruturas. Contém as características do espaço, nomeadamente:

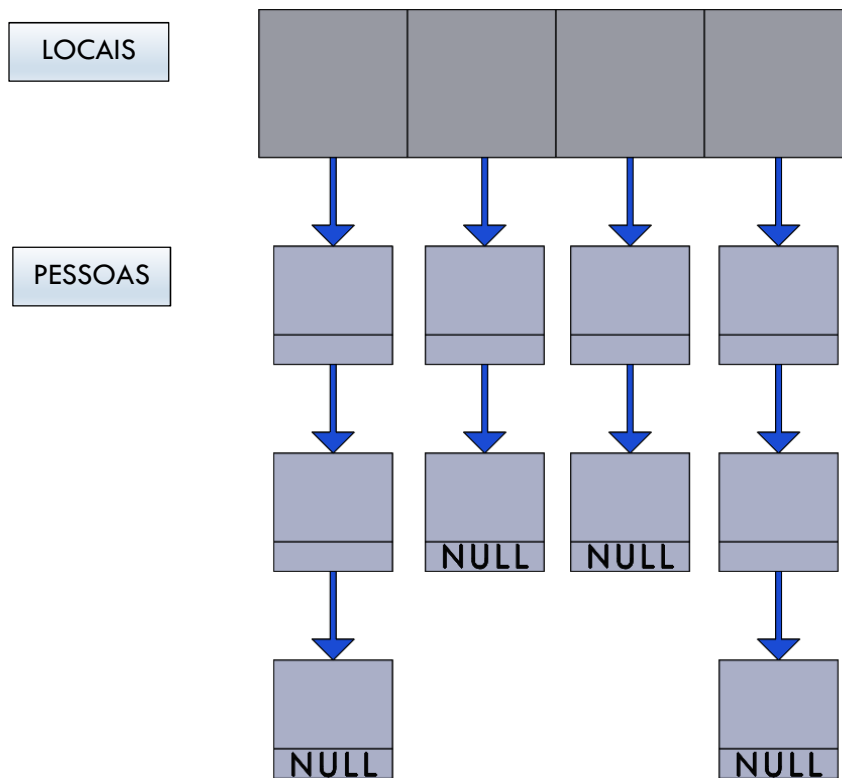
- id, identificador do local, único e positivo;
- capacidade, que é a capacidade local máxima;
- liga[3], que tem as ligações dos locais, máx. de 3.

```
typedef struct salaaux localaux,
*plocalaux;
struct salaaux{
    int id;
    int capacidade;
    int liga[3];
    int contador;
    pno pess;
    pno p1;
    pno p2;
    pno p3;
};
```

A estrutura “localaux”, que foi utilizada no decorrer do trabalho, é em parte semelhante à estrutura “local”, no entanto tem mais algumas características:

- contador, que contabiliza a quantidade de pessoas que se encontram em determinado local;
- pno pess, que é um ponteiro para pessoa, e que vai apontar para a lista atual de pessoas;
- pno p1, ponteiro para pessoa que aponta para a lista de pessoas de uma iteração anterior;
- pno p2, ponteiro para pessoa que aponta para a lista de pessoas de duas iterações anteriores;
- pno p3, ponteiro para pessoa que aponta para a lista de pessoas de três iterações anteriores;
- \*plocalaux, ponteiro para estrutura localaux.

# ESTRUTURAS DINÂMICAS



A informação do espaço, que contém os vários locais, encontra-se armazenada num vetor dinâmico de estruturas de dados do tipo “localaux”, apresentadas anteriormente. As pessoas em estudo foram distribuídas aleatoriamente pelos vários locais, e a informação destas pessoas está armazenada em estruturas dinâmicas do tipo lista ligada. Associar as pessoas aos locais onde se encontram, no vetor, foi a opção tomada de forma a melhorar a organização e a otimizar a execução do programa. Assim, quando se pretende trabalhar na informação de pessoas que se encontram num determinado local, não é necessário percorrer uma lista completa à procura das pessoas que estão nesse local, pois, com este método, depois de se encontrar o local, tem-se acesso à lista completa das pessoas que lá se encontram, ganhando tempo de simulação.

## IMPLEMENTAÇÃO DO PROGRAMA

Neste tópico serão abordadas as principais opções tomadas ao nível da implementação e a sua justificação. Como já foi visto no tópico anterior, a lista de pessoas ficou associada ao local onde se encontravam no vetor, sendo esta a lógica seguida no trabalho. O trabalho consistiu em duas fases principais, a de preparação dos dados, e a fase de simulação.

### FASE DE PREPARAÇÃO

No início da execução do programa, os dados do espaço e dimensão e os dados das pessoas são carregados a partir de ficheiros.

#### 1. ESPAÇO

O carregamento das configurações do espaço foi feito a partir de um ficheiro binário para um vetor dinâmico.

Para isso, pediu-se o nome do ficheiro ao utilizador, que tem obrigatoriamente que ter a extensão “.bin”. Caso o utilizador insira um nome inválido, ou o ficheiro esteja inacessível, tem a opção de inserir novamente o nome do ficheiro quantas vezes quiser, se assim pretender. Se optar por não introduzir mais nenhuma vez o nome do ficheiro, a fase de preparação é interrompida e a simulação não é iniciada.

Se o utilizador inserir um nome válido, é aberto o ficheiro binário e a estrutura do tipo “local” é utilizada unicamente para ler os dados do ficheiro binário, que estão nesse formato. Enquanto estes dados forem lidos, são guardados num vetor que armazena estruturas do tipo “localaux”, tantas quanto o número de locais que forem lidos.

Os locais que foram carregados para o vetor são então mostrados ao utilizador, e estão agora prontos a ser validados. A função de validação vai analisar se são cumpridas cinco condições: 1. Os ID's devem ser únicos. 2. Os ID's devem ser positivos. 3. O ID do local não pode estar repetido no array liga, isto é, o local não pode estar ligado a si próprio. 4. As ligações no array liga não podem estar duplicadas, excetuando o caso “-1” (por exemplo, não pode existir a ligação [4 4 -1], pois a informação sobre a ligação ao ID 4 está repetida, mas pode existir a ligação [4 -1 -1]). 5. As ligações devem ser recíprocas, isto é, se o local 3 está ligado ao local 1, o local 1 também deve ter ligação ao local 3.

Se todas as validações estiverem corretas, o programa avança para a segunda fase de preparação. Caso contrário, a execução é interrompida sem passar pela fase de simulação.

#### 2. PESSOAS

O carregamento das configurações das pessoas começou por ser feito a partir de um ficheiro txt para uma lista ligada única.

Para isso, pediu-se o nome do ficheiro ao utilizador, que tem obrigatoriamente que ter a extensão “.txt”. Caso o utilizador insira um nome inválido, ou o ficheiro esteja inacessível, tem a opção de inserir novamente o nome do ficheiro quantas vezes quiser, se assim pretender. Se optar por não introduzir mais nenhuma vez o nome do ficheiro, a fase de preparação é interrompida e a simulação não é iniciada.

Se o utilizador inserir um nome válido, é aberto o ficheiro txt e a estrutura do tipo “pessoas” é utilizada para ler os dados do ficheiro txt, que estão nesse formato. Enquanto estes dados forem lidos, tendo em atenção que só são lidos até ao máximo da capacidade total do espaço, e se existirem mais pessoas as mesmas são ignoradas, os dados são alocados num novo nó do tamanho da estrutura “pessoas”. Os novos

nós vão sendo colocados na cabeça da lista, por ser a forma mais simples e não ser requerido nenhum tipo de ordenação.

As pessoas que foram carregadas para a lista são então mostradas ao utilizador, e estão agora prontas a ser validadas. A função de validação vai analisar se não existem pessoas com o mesmo ID.

Se a validação estiver correta, as pessoas são distribuídas pelos vários locais, de forma aleatória, satisfazendo a restrição da capacidade do local. Para isso, os nós da lista única vão sendo distribuídos para o ponteiro de pessoa que está na estrutura do local com o respetivo ID. Caso contrário, a execução é interrompida sem passar pela fase de simulação.

Finalmente, depois de cumpridas estas etapas, o programa avança para a fase de simulação.

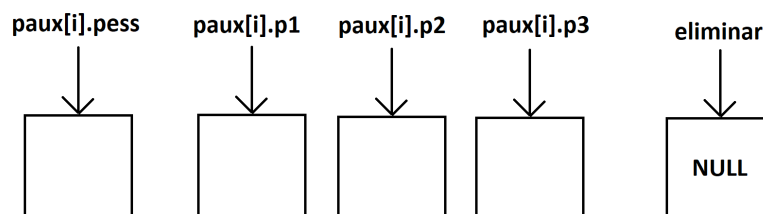
## FASE DE SIMULAÇÃO

Na fase de simulação, o utilizador tem acesso a um Menu com sete opções:

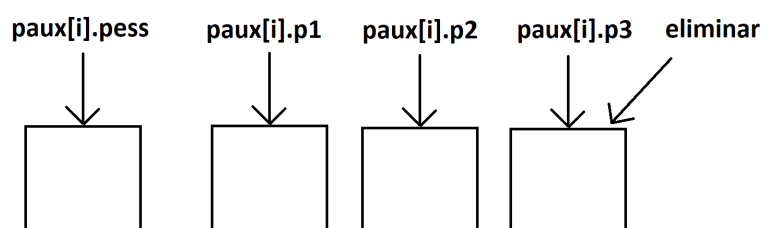
1. Avançar Iteração
2. Apresentar Estatísticas
3. Adicionar Doente
4. Transferir Pessoas
5. Listar Pessoas
6. Retroceder Iterações
7. Terminar Simulação

Antes de explicar o que é feito em cada opção do menu, é importante mencionar que existe um contador de iterações, e uma função que antes de cada iteração ser concretizada (nomeadamente: Avançar Iteração, Adicionar Doente, ou Transferir pessoas), assegura que os ponteiros P1, P2, P3 sejam atualizados. Como já foi mencionado na parte das Estruturas de Dados, as estruturas do tipo “localaux” contêm quatro ponteiros de pessoas, um que guarda a lista atual, e três que guardam as listas de até três iterações anteriores. Assim, quando o utilizador escolher a opção “6. Retroceder Iterações”, é possível retroceder o número de iterações pretendido.

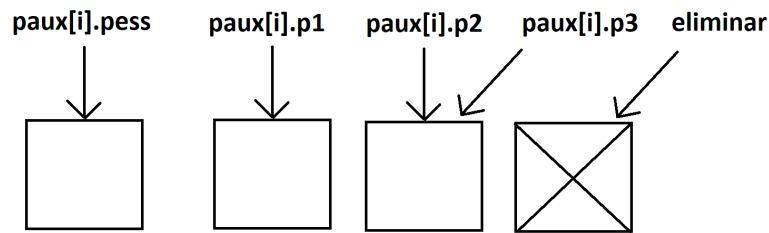
A função “atualizalistasparaiteracoes” será assim chamada no instante imediatamente anterior a ser concretizada uma nova iteração, e esquematicamente faz o seguinte:



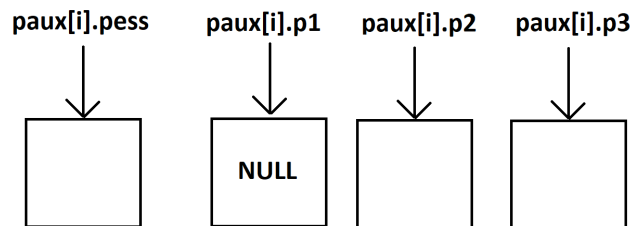
Começa por colocar o ponteiro para pessoa “eliminar” a apontar para a lista p3:



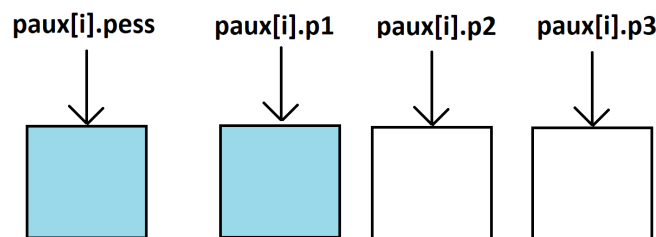
Depois o ponteiro da lista p3 passa a apontar para a lista p2, e elimina a lista p3:



Seguidamente, o ponteiro da lista P2 passa a apontar para a lista P1, e o ponteiro P1 passa a apontar para NULL:



Por fim, uma cópia da lista atual é feita para P1, e assim a lista atual está pronta a ser atualizada. Desta forma, garante-se que os dados anteriores às iterações são sempre salvaguardados, e o utilizador pode retroceder as iterações que pretender:



Uma vez explicado o funcionamento desta função, seguidamente quando se detalharem as opções do Menu, será apenas mencionada a chamada a esta função e não será novamente explicado o seu funcionamento.

### 1. AVANÇAR ITERAÇÃO

Assim que o utilizador escolhe a opção 1, é feita a chamada à função de atualizar as listas para as iterações, e o número de iterações é incrementado.

Esta função de Avançar Iteração divide-se em duas fases: na primeira fase são incrementados dias às pessoas doentes até um máximo de dias permitido, passando a Saudáveis depois de ultrapassado o limite de dias, ou, ainda, as pessoas doentes podem ter uma recuperação espontânea, cuja probabilidade depende da idade da pessoa. No caso de a pessoa ficar curada, quer seja por já terem passado os dias permitidos, ou por ter recuperado espontaneamente, é calculada a probabilidade de ficar Imune à doença, e dependendo do resultado, o seu estado passa a “Saudável” ou a “Imune”.

Depois de curadas as pessoas, é aplicada a taxa de disseminação, em que um doente pode infetar 5% das pessoas que se encontram na sala, escolhidas de forma aleatória.

Esta foi a maneira escolhida para Avançar Iteração, pois acredito que esta abordagem de curar primeiro as pessoas e depois aplicar a taxa de disseminação faz mais sentido. No entanto, com esta abordagem, é possível que uma pessoa doente que tenha ficado curada numa primeira fase, passando a saudável, ao ser aplicada a taxa de disseminação possa ser infetada novamente.

## 2. APRESENTA ESTATÍSTICAS

A função de Apresenta Estatísticas mostra a distribuição das pessoas por salas, contabiliza o número de doentes, saudáveis e infetados em cada sala, além de mostrar o ID do local, a capacidade, e a quantidade total de pessoas que se encontram lá. No fim, apresenta ainda o número total de doentes, o número total de infetados, e o número total de pessoas saudáveis, bem como as respetivas percentagens.

## 3. ADICIONAR DOENTE

Quando o utilizador escolhe a opção 3, é-lhe questionado o ID do local onde pretende inserir o novo doente, e a função vai verificar se esse ID existe e se tem capacidade para ser lá inserida uma nova pessoa. Depois, é alocado um novo nó do tipo “pessoas”, e o utilizador deve inserir o nome do doente, sendo chamada uma função que verifica se esse nome já existe. No caso de existir, não é possível inserir o novo doente e a função termina. Pelo contrário, se o nome não existir, o utilizador insere a idade e o número de dias da infeção. Depois de todos os dados preenchidos no novo nó, é feita a chamada da função de atualizar as listas para as iterações, e o número de iterações é incrementado. Por fim, o novo nó é inserido na cabeça da lista das pessoas que se encontram nesse local, e o contador de pessoas do local bem como o número total de pessoas é incrementado.

## 4. TRANSFERIR PESSOAS

Ao escolher a opção 4, o utilizador insere o número de pessoas a transferir, o ID do local de origem e o ID do local de destino. Depois, é feita a chamada a uma função que verifica se são cumpridas as condições de transferência, nomeadamente: 1. Procura o ID do local de origem, se o encontrar verifica a quantidade de pessoas que tem, que deve ser pelo menos igual à quantidade de pessoas a transferir, e se o local de origem tem ligação direta ao local de destino; 2. Procura o ID do local de destino, e assegura que o número de pessoas que tem acrescido do número de pessoas a transferir é menor ou igual à sua capacidade total, e verifica que tem ligação direta ao local de origem.

Se as condições de transferência forem cumpridas, é feita a chamada à função de atualizar as listas para as iterações, e o número de iterações é incrementado. Caso contrário, não é possível transferir pessoas e a função retorna ao menu.

Por fim, é efetuada a transferência das pessoas, escolhendo aleatoriamente o número de pessoas a transferir do local de origem, entre as que se encontram no espaço, e a transferência é feita pessoa a pessoa, sendo a pessoa inserida na cabeça da lista do local de destino.

## 5. LISTAR PESSOAS

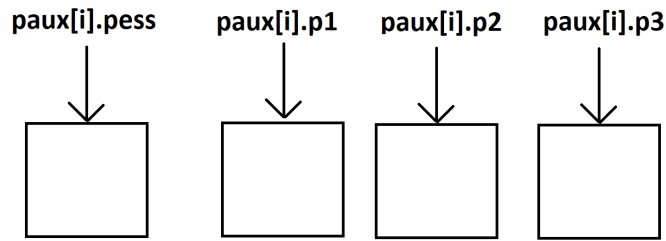
Esta função foi colocada de forma a tornar mais prático visualizar a distribuição das pessoas por sala e o estado em que se encontram após realizar alguma iteração. Na prática apresenta a lista das pessoas que se encontram em determinado local.

## 6. RETROCEDER ITERAÇÕES

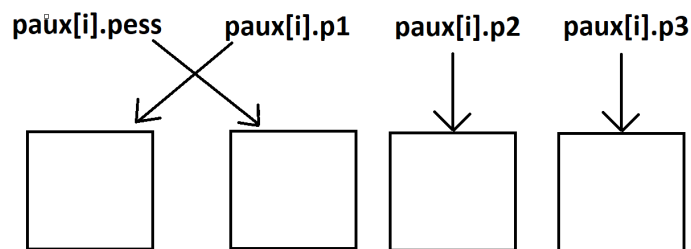
Para retroceder iterações, até um máximo de três, é questionado ao utilizador o número de iterações que deseja retroceder. Se o número inserido for superior ao número indicado pelo contador de iterações, é apresentada uma mensagem de que não é possível retroceder, e a função retorna ao menu.

No caso de ser possível retroceder o número de iterações inserido, e uma vez que as listas estão atualizadas devido à função de atualizar as listas para as iterações, agora só é necessário fazer uma troca de ponteiros.

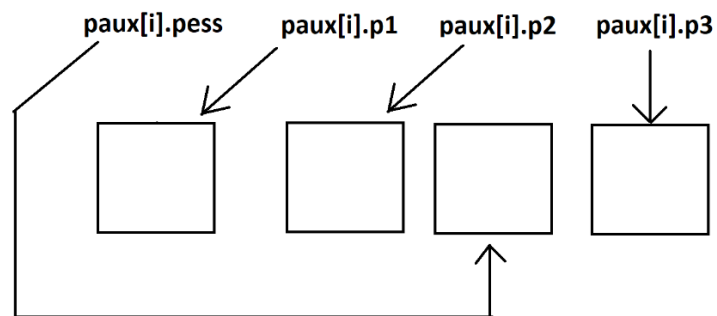




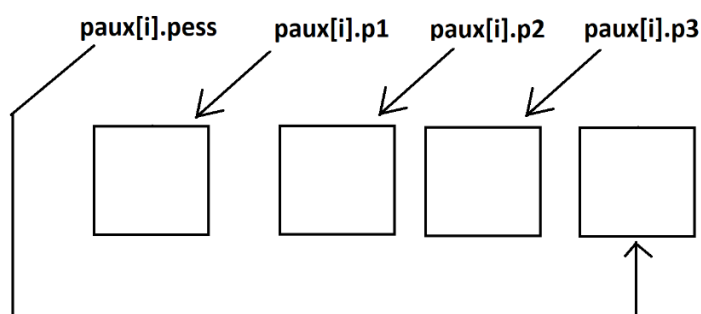
Se o utilizador pretender **retroceder uma iteração**, o ponteiro da lista atual passa a apontar para o ponteiro da lista p1, e vice-versa, recorrendo para isso a um ponteiro auxiliar, que permite que não se perca algum ponteiro. Desta forma, o estado atual da iteração não é perdido, e se o utilizador selecionar novamente para retroceder uma iteração, volta ao estado imediatamente anterior. No final este é o resultado:



Se o utilizador pretender **retroceder duas iterações**, recorrendo também a um ponteiro auxiliar que permite que não se perca nenhum ponteiro, a troca a efetuar é que o ponteiro P2 passa a apontar para P1, o ponteiro P1 passa a apontar para a lista Atual, e o ponteiro da lista Atual passa a apontar para P2:



Por fim, se o utilizador pretender **retroceder três iterações**, recorrendo também a um ponteiro auxiliar que permite que não se perca nenhum ponteiro, a troca a efetuar é que o ponteiro P3 passa a apontar para P2, o ponteiro P2 passa a apontar para P1, o ponteiro P1 passa a apontar para a lista Atual, e o ponteiro da lista Atual passa a apontar para P3:



Por fim, o contador das pessoas nos locais é atualizado, de forma a repor o estado da iteração anterior.

Esta foi a forma implementada para retroceder iterações pois considerei que era também a mais simples, já que com a função de atualizar as listas para iterações, realizada imediatamente antes da iteração ser concretizada, garanto sempre que as listas estão atualizadas a cada nova iteração realizada, e depois é só fazer uma troca de ponteiros para repor o estado anterior do programa.

## 7. TERMINAR SIMULAÇÃO

Ao terminar a simulação, a primeira etapa consiste em guardar a lista atual de pessoas num ficheiro txt, cujo nome é dado pelo utilizador, e que deve ter obrigatoriamente a extensão pretendida. O ficheiro é assim guardado no formato pretendido, salvaguardando que as pessoas saudáveis ou imunes não têm informação de dias de infeção.

Depois, é elaborado um relatório para um ficheiro txt com o nome já pré-definido, e a informação colocada neste relatório é praticamente a mesma informação que foi feita para as Estatísticas, pois considerei que já estava bastante completa.

Por fim, são eliminadas as listas e libertado o vetor, uma vez que já não é necessário.

# MANUAL DE UTILIZAÇÃO

## FASE DE PREPARAÇÃO

### 1. ESPAÇO

Inserir o nome do ficheiro Espaço pretendido.

```

trabalho_sandra_2020
***** SIMULACAO DA PROPAGACAO DE VIRUS COVID-19 *****
*
*                               ISEC - Programacao 2019/2020
*
*                               Aluna: Sandra Perdigao Nr: 2019102697
*
*                               Docente Responsavel: Professor Francisco Pereira
*
*****
*****

Insira o nome do ficheiro de espaco a abrir(ex: E1.bin):E1.bin

LOCAIS CARREGADOS:

ID: 1  Capacidade: 50  Ligacoes: 3 -1 -1
ID: 2  Capacidade: 50  Ligacoes: 3 -1 -1
ID: 3  Capacidade: 50  Ligacoes: 1 4 2
ID: 4  Capacidade: 50  Ligacoes: 3 -1 -1

Locais validados: a 1a fase de preparacao foi concluida com sucesso.
*****

```

### 2. PESSOAS

Inserir o nome do ficheiro Pessoas pretendido.

```

Insira o nome do ficheiro de pessoas a abrir(ex: pessoasA.txt):pessoasB.txt

PESSOAS CARREGADAS:

LeonardoL      10      S
TobiasF 98      S
DianaD 89      S
PedroPedra     27      S
SandraMotaA1   24      S
CarlaCardoso   34      D      7
Carlos1 20      D      3
PauloAlves123  34      D      4
HugoC 14      D      5
HugoB 25      D      10
HugoA 20      D      3
SandraS 56      D      1
SamuelSimoes   56      D      1
Olivia1 12      S
Zulmira2A      17      S
MarcoMarquesAAA 40      I
PauloPires2    67      D      10
Tomas111       12      I
SofiaA 23      S
PauloPires1    23      S

Pessoas validadas: a 2a fase de preparacao foi concluida com sucesso.
*****

```

## FASE DE SIMULAÇÃO

Escolher um número de entre as várias opções que listam no Menu:

```
***** FASE DE SIMULACAO *****
1 - Avanca Iteracao:
2 - Apresenta Estatistica:
3 - Adiciona Doente:
4 - Transferir Pessoas
5 - Lista Pessoas
6 - Retroceder Iteracoes
7 - Terminar Simulacao
```

### 1. AVANÇAR ITERAÇÃO

Para ver resultado pode escolher a opção 5 – Lista pessoas.

```
*****
LISTA DE PESSOAS NOS DIFERENTES LOCAIS:

LOCAL ID: 1      CAPACIDADE: 50  CONTADOR:6
PauloPires1     23      S
Tomas111        12      I
MarcoMarquesAAA 40      I
PauloAlves123    34      D      5
Carlos1 20      D      4
CarlaCardoso     34      D      8

LOCAL ID: 2      CAPACIDADE: 50  CONTADOR:4
SofiaA  23      S
HugoB   25      S
TobiasF 98      S
LeonardoL 10     S

LOCAL ID: 3      CAPACIDADE: 50  CONTADOR:5
PauloPires2     67      D      11
Zulmira2A       17      S
HugoA   20      D      4
HugoC   14      D      6
PedroPedra      27      S

LOCAL ID: 4      CAPACIDADE: 50  CONTADOR:5
Olivia1 12      S
SamuelSimoes    56      D      2
SandraS  56      D      2
SandraMotaA1    24      S
DianaD   89      S

*****
```

### 2. APRESENTAR ESTATÍSTICA

Apresenta alguns dados, sendo esta uma pequena amostra:

```
-----
LOCAL 4 CAPACIDADE-50 pessoas  OCUPACAO-5 pessoas
Olivia1 12      S
SamuelSimoes    56      D      2
SandraS  56      D      2
SandraMotaA1    24      S
DianaD   89      S

Pessoas Doentes: 2
Pessoas saudaveis: 3
Pessoas imunes: 0

-----

TOTAL PESSOAS DOENTES: 8      40.00 pct
TOTAL PESSOAS SAUDAVEIS: 10    50.00 pct
TOTAL PESSOAS IMUNES: 2 10.00 pct

***** FIM ESTATISTICAS *****
```

## 3. ADICIONAR DOENTE

```

3
Insira o ID do local: 2
Insira o nome do doente: aNA
Insira a idade do doente: 40
Insira o numero de dias da infecao: 2

```

## 4. TRANSFERIR PESSOAS

```

4
Insira a quantidade de pessoas a mover: 2
Insira o ID de origem: 2
Insira o ID de destino: 3

```

## 6. RETROCEDER ITERAÇÕES

```

6
Introduza o numero de iteracoes que pretende retroceder: 1

```

A qualquer momento pode-se consultar a Lista de Pessoas com a opção 5:

```

*****
LISTA DE PESSOAS NOS DIFERENTES LOCAIS:

LOCAL ID: 1      CAPACIDADE: 50  CONTADOR:6
PauloPires1     23      S
Tomas111        12      I
MarcoMarquesAAA 40      I
PauloAlves123   34      D      5
Carlos1 20      D      4
CarlaCardoso    34      D      8

LOCAL ID: 2      CAPACIDADE: 50  CONTADOR:5
aNA 40          D      2
SofiaA 23       S
HugoB 25       S
TobiasF 98      S
LeonardoL 10    S

LOCAL ID: 3      CAPACIDADE: 50  CONTADOR:5
PauloPires2     67      D      11
Zulmira2A       17      S
HugoA 20        D      4
HugoC 14        D      6
PedroPedra      27      S

LOCAL ID: 4      CAPACIDADE: 50  CONTADOR:5
Olivia1 12      S
SamuelSimoes    56      D      2
SandraS 56      D      2
SandraMotaA1    24      S
DianaD 89       S
*****

```



## 7. TERMINAR SIMULAÇÃO

```

7
Introduza o nome do ficheiro txt (Ex: pessoas.txt): pessoas.txt
As pessoas foram corretamente guardadas no ficheiro pessoas.txt.
Relatorio Elaborado com sucesso no ficheiro report.txt. Ate a proxima!

***** FIM DA SIMULACAO *****

```

 pessoas	14/06/2020 21:54	Documento de texto	1 KB
 report	14/06/2020 21:54	Documento de texto	2 KB