

Dispositivo para gerenciamento de comprimidos

Mateus Alves da Rocha

Engenharia Eletrônica

Universidade de Brasília-UnB/FGA Gama, DF.

e-mail: mateus.alves.unb@gmail.com

Mayara Barbosa dos Santos

Engenharia Eletrônica

Universidade de Brasília-UnB/FGA Gama, DF.

e-mail: mayara.b97@gmail.com

Abstract—Com o advento dos vários problemas de saúde, as pessoas vêm-se obrigadas a tomar vários remédios. Com o intuito de ajudar os usuários para que não esqueçam de ingerir seus medicamentos e acarretarem problemas mais sérios por falta de seu uso, este projeto tem o propósito de auxiliá-los no dia-a-dia e um dispositivo para gerenciamento de comprimidos será desenvolvido por intermédio do microcontrolador *Raspberry Pi*.

Palavras-chave - Gerenciamento, comprimidos, *Raspberry Pi*

I. INTRODUÇÃO

Sabe-se que, com o envelhecimento, há uma tendência de diminuição na capacidade de memorização de um indivíduo. Isto é agravado, caso esta pessoa sofra de alguma doença que afete diretamente nessa habilidade.[1]

Entretanto, não é incomum que pessoas esqueçam coisas importantes independente de enfermidades relacionadas à perda de memória ou o envelhecimento. O desenvolvimento tecnológico pode ter desempenhado um papel neste problema, dado que desde que os *smartphones* facilitaram o uso de agendas para contatos é fácil encontrar alguém que não tenha memorizado o próprio número de sua residência, por exemplo. Apesar disso, o esquecimento de contatos ou termos que podem ser facilmente encontrados com poucos minutos de pesquisa na internet não constitui um problema sério para a população. Porém, um dos problemas da falta de exercício no sentido de melhorar a capacidade de memorização é a dificuldade em seguir prescrições médicas de medicamentos.

Os idosos são os mais afetados por este problema. Uma pesquisa realizada na Universidade Estadual de Campinas e publicada na revista *Ciências e Saúde Coletiva* entrevistou 165 idosos e constatou que 58,2% possui acima de quatro comorbidades simultâneas o que leva a um número considerável de remédio para gerenciar ao longo do dia.[2] E, neste mesmo estudo, 55,2% afirmou não ter cuidador. A necessidade de vários comprimidos diariamente e em horários diferentes é dificultado pelo esquecimento, trabalho e déficit cognitivo.[3]

No entanto, como mencionado anteriormente, não são apenas idosos que possuem dificuldades relacionadas ao número simultâneo de medicamentos. Kourrouski e Lima publicaram um estudo na revista *Latino Americana de*

Enfermagem que apesar de os adolescentes diagnosticados como portadores do HIV relatarem saber dos benefícios da medicação no controle uma grande parcela deles não adere aos tratamentos por diversos fatores e dentre eles inclui-se o esquecimento do medicamento. Essas pesquisadoras afirmam ainda que é necessário orientá-los para uso de despertadores para que não esqueçam o horário correto das medicações. [4]

Há sistemas comerciais desenvolvidos voltados ao gerenciamento de medicamentos. Como os sistemas das imagens abaixo. Entretanto, os preços dessas tecnologias ainda estão pouco acessíveis a grande parte da população. O produto *MedFolio Wireless Pillbox* modelo WP1050 custa \$250,95 de acordo com o site da *Amazon* [5]. Já o *MedMinder Maya* funciona a partir de assinatura que variam de \$40 a \$60 por mês. [6]

Existem também alguns sistemas amadores que buscam atender essa demanda. Um exemplo é o dispositivo de Wojtek Siudzinski que utiliza um servo motores para fazer a movimentação de discos impressos em uma impressora 3D para dispensar pílulas de MM, mas pode ser utilizado também para comprimidos. [7]

Thomas Nabelek e Adam Nolte, alunos da Universidade de Missouri nos Estados Unidos desenvolveram um projeto que busca automatizar o gerenciamento de comprimidos. O sistema prevê o controle inclusive de farmacêuticos através da possibilidade de acompanhamento da rotina de remédios através da Web. [8]

Neste cenário, o sistema desenvolvido neste trabalho traz uma solução tecnológica para o controle de medicamentos. Busca-se retirar dos pacientes a responsabilidade desse gerenciamento e ao mesmo tempo garantir uma alta confiabilidade que os medicamentos serão lembrados e administrados conforme prescritos pelos profissionais da saúde.

II. OBJETIVOS

O objetivo do projeto é desenvolver um sistema que auxilie o usuário ingerir seus comprimidos corretamente de forma que facilite sua rotina e não interrompa seu tratamento. O produto em questão terá um banco de dados com todos os usuários que serão cadastrados assim como

todas as informações pertinentes para que o usuário insira o remédio adequadamente, o usuário terá fácil acesso ao sistema por meio de reconhecimento facial.

III. REQUISITOS

Utilizando o hardware Raspberry Pi que comporta diversas distribuições Linux como plataforma de desenvolvimento do produto proposto ao projeto tem os seguintes requisitos:

- Permitir ao administrador do sistema configurar a rotina de horários e quantidade de remédios a serem prescritos. Além de ser possível cadastrar os usuários do sistema que será armazenado em um banco de dados. Para este fim, haverá uma interface gráfica facilitando a interação com o usuário;
- Emitir no horário configurado um aviso por meio de um *buzzer* para o usuário do sistema tomar o remédio na hora certa;
- Identificar o usuário e associar a ele a rotina de administração de remédios específica por meio de reconhecimento facial a partir de uma câmera que fará a comunicação com a Raspberry Pi;
- Apresentar um dispositivo eletromecânico para dispensar o remédio automaticamente. O mesmo vai operar com motor de passo e um servo motor controlado pela Raspberry Pi.

IV. BENEFÍCIOS

O projeto apresenta os seguintes benefícios:

- 1) O usuário será lembrado da hora que terá que ingerir o remédio;
- 2) O tratamento da doença a ser tratada não será interrompido;
- 3) Evitar problemas mais sérios nos casos de doenças crônicas;
- 4) Baixo custo.

REFERÊNCIAS

- [1] O. P. Almeida, "Queixa de problemas com a memória e o diagnóstico de demência," Arq. Neuropsiquiatr., vol. 56, no. 3 A, pp. 412–418, 1998.
- [2] F. A. Cintra, M. E. Guariento, and L. A. Miyasaki, "Adesão medicamentosa em idosos em seguimento ambulatorial," Cien. Saude Colet., vol. 15, pp. 3507–3515, 2010.
- [3] Teixeira JJV, Spínola AWP. Comportamento do paciente idoso frente à aderência medicamentosa. Arq Geriatr Gerontol 1998; 2(1):5-9.
- [4] M. F. C. Kourouski and R. A. G. de Lima, "Adesão ao tratamento: vivências de adolescentes com hiv/aids," Rev Latino-am Enferm., vol. 17, no. 6, p. 111, 2009.
- [5] AMAZON, "MedFolio Wireless Pillbox (WP1050)." [Online]. Available: <https://www.amazon.com/Medfolio-MedFolio-Wireless-Pillbox-WP1050/dp/B00D3B7TVQ>.
- [6] The Sweethome, "The best smart pill dispenser (so far)." [Online]. Available: <https://www.engadget.com/2017/09/10/the-best-smart-pill-dispenser-so-far/>.
- [7] W. Siudzinzi, "Spark Core powered automatic pill dispenser," 2014. [Online]. Available: <https://suda.pl/spark-core-powered-automatic-pill-dispenser/>.
- [8] T. Nabelek and A. Nolte, "Automatic Pill Dispenser," 2016. [Online]. Available: <https://www.youtube.com/watch?v=0gYyqYY8B-M>.

A. APÊNDICE

```
1
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <string.h>
5
6 char usuario[10][40];
7 int idade[10];
8 int quantidade_remedio[10][1];
9 char remedio[10][10][20];
10 int quantidade_dose[10][10][1];
11 int quantidade_horario[10][10][1];
12 int hora[10][10][3];
13 int minuto[10][10][3];
14
15 char hora_lida1[1];
16 char minuto_lida1[1];
17 char hora_lida2[1];
18 char minuto_lida2[1];
19 char hora_lida4[1];
20 char minuto_lida4[1];
21 char idade_lida[3];
22 char remedio_lida[3];
23 char dose_lida[3];
24
25 char leitura[10][400];
26 char *result;
27
28 int qtdp=0;
29 int r_qtdp=-1;
30 int i=0;
31 int j=0;
32 int a=0;
33 int selecao=0;
34 int inicializacao=0;
35
36 int escrever_dados();
37 int ler_dados();
38 int mostrar_dados();
39 int limpar_dados();
40 int chamar_codigo(int selecao);
41
42 //Criar ponteiro de arquivo
43 FILE *pont_arq;
44
45 int main(){
46
47
48 //abre arquivo
49 pont_arq=fopen("teste3.txt","r");
50
51 if (pont_arq == NULL ) {
52     printf("\n**** arquivo nao existe ****\n");
53     pont_arq=fopen("teste3.txt","w");
54
55 //cabecalho
56 fprintf(pont_arq, "%s", "usuario");
57 fprintf(pont_arq, "\t");
58 fprintf(pont_arq, "%s", "idade");
59 fprintf(pont_arq, "\t");
60 fprintf(pont_arq, "%s", "Quantidade de remedios");
61 fprintf(pont_arq, "\t");
62 fprintf(pont_arq, "%s", "nome dos remedios");
63 fprintf(pont_arq, "\t");
64 fprintf(pont_arq, "%s", "Quantidade de dose");
65 fprintf(pont_arq, "\t");
66 fprintf(pont_arq, "%s", "horario 01");
67 fprintf(pont_arq, "\t");
68 fprintf(pont_arq, "%s", "horario 02");
69 fprintf(pont_arq, "\t");
70 fprintf(pont_arq, "%s", "horario 03");
71 fprintf(pont_arq, "\n");
72 }
```

```

73 }else{
74     printf("\n****  arquivo aberto com sucesso
75     ****\n");
76     ler_dados();
77     inicializacao=1;
78 };
79 fclose(pont_arq);
80
81 while (selecao!=5){
82     while (selecao<1 || selecao>5){
83         printf("\n****  Qual modo de operacao deseja
84         ****\n");
85         printf("\n****  Ler banco de dados : 1 ****\n
86         ");
87         printf("\n****  cadastrar dados : 2 ****\n");
88         printf("\n****  Mostrar dados : 3 ****\n");
89         printf("\n****  Limpar banco de dados : 4
90         ****\n");
91         printf("\n****  Fechar programa : 5 ****\n");
92         scanf("%d", &selecao);
93     };
94     printf("\nModo selecionado : %d\n", selecao);
95     chamar_codigo(selecao);
96     fclose(pont_arq);
97
98     if (selecao<5 && selecao!=0){
99         selecao=0;
100     };
101 };
102
103
104 };
105
106
107 fclose(pont_arq);
108 printf("\nARQUIVO FECHADO\n");
109 getchar();
110
111 return (0);
112 }
113
114 int escrever_dados(){
115
116     int qtd=0;
117     int mem;
118
119     pont_arq=fopen("teste3.txt","w");
120
121     // *****
122
123     //prepara cabecalho do banco de dados
124     fprintf(pont_arq, "%s", "usuario");
125     fprintf(pont_arq, "\t");
126     fprintf(pont_arq, "%s", "idade");
127     fprintf(pont_arq, "\t");
128     fprintf(pont_arq, "%s", "Quantidade de remedios
129     ");
130     fprintf(pont_arq, "\t");
131     fprintf(pont_arq, "%s", "nome dos remedios");
132     fprintf(pont_arq, "\t");
133     fprintf(pont_arq, "%s", "Quantidade de dose");
134     fprintf(pont_arq, "\t");
135     fprintf(pont_arq, "%s", "horario 01");
136     fprintf(pont_arq, "\t");
137     fprintf(pont_arq, "%s", "horario 02");
138     fprintf(pont_arq, "\t");
139     fprintf(pont_arq, "%s", "horario 03");
140     fprintf(pont_arq, "\n");
141
142     //

```

```

142 *****
143
144 //
145 *****
146 //Inicio do cadastro de pacientes
147
148 printf("\nEntre com a quantidade de pacientes: ")
149 ;
150 scanf("%d", &qtd);
151 getchar();
152 mem=qtdp;
153 qtdp=qtd+qtdp;
154
155 for (i=0;i<qtdp;i++){ //i = numero do paciente
156
157     //
158     *****
159
160     //ZERAR VARIAVEIS TIPO CONTADORES DAS LOGICAS
161     DE CADA USUARIO
162     j=0;
163     a=0;
164     //
165     *****
166
167     //Nome do usuario
168     if(i==mem || inicializacao<1){ //garante a nao
169         escrita sobre o dado anterior
170         printf("\nEntre com o nome do usuario %d: ",
171             (i+1));
172         gets(usuario[i]);

```

```

168     };
169     fprintf(pont_arq, "%s", usuario[i]);
170     fprintf(pont_arq, "\t");
171
172     //Idade do usuario
173     if(i==mem || inicializacao<1){ //garante a nao
174         escrita sobre o dado anterior
175         printf("\nEntre com a idade do usuario %d: ",
176             (i+1));
177         scanf("%d", &idade[i]);getchar();
178     };
179     fprintf(pont_arq, "%d", idade[i]);
180     fprintf(pont_arq, "\t");
181
182     //Quantidade de remedio
183     if(i==mem || inicializacao<1){ //garante a nao
184         escrita sobre o dado anterior
185         printf("\nEntre com a quantidade de remedios
186         do usuario %d: ", (i+1));
187         scanf("%d", &quantidade_remedio[i][0]);
188         getchar();
189     };
190     fprintf(pont_arq, "%d", quantidade_remedio[i
191         ][0]);
192     fprintf(pont_arq, "\t");
193
194     for(j=0;j<quantidade_remedio[i][0];j++){ //laco
195         do remedio
196
197         //Nome do remedio
198         if(i==mem || inicializacao<1){ //garante a
199             nao escrita sobre o dado anterior
200             printf("\nEntre com o nome do remedio %d do
201             usuario %d: ", (j+1), (i+1));
202             gets(remedio[i][j]);
203         };

```

```

195     fprintf(pont_arq, "%s", remedio[i][j]);
196     fprintf(pont_arq, "\t");
197
198     //Quantidade de dose do remedio
199     if(i==mem || inicializacao<1){ //garante a
nao escrita sobre o dado anterior
200         printf("\nEntre com a quantidade de dose do
remedio %d do usuario %d: ", (j+1), (i+1));
201         scanf("%d", &quantidade_dose[i][j][0]);
getchar();
202     };
203     fprintf(pont_arq, "%d", quantidade_dose[i][j]
][0]);
204     fprintf(pont_arq, "\t");
205
206     //Quantidade de horarios do remedio
207     if(i==mem || inicializacao<1){ //garante a
nao escrita sobre o dado anterior
208         printf("\nEntre com a quantidade de
horarios do remedio %d do usuario %d: ", (j+1),
(i+1));
209         scanf("%d", &quantidade_horario[i][j][0]);
getchar();
210     };
211
212     for(a=0;a<quantidade_horario[i][j][0];a++){
213         //Hora do remedio
214         if(i==mem || inicializacao<1){ //garante a
nao escrita sobre o dado anterior
215             printf("\nEntre com a hora %d do remedio
%d do usuario %d: ", (a+1), (j+1), (i+1));
216             scanf("%d", &hora[i][j][a]);getchar();
217         };
218
219         //Minuto do remedio
220         if(i==mem || inicializacao<1){ //garante a
nao escrita sobre o dado anterior
221             printf("\nEntre com o minuto %d do
remedio %d do usuario %d: ", (a+1), (j+1), (i
+1));
222             scanf("%d", &minuto[i][j][a]);getchar();
223         };
224             fprintf(pont_arq, "%d:%d", hora[i][j][a],
minuto[i][j][a]);
225             fprintf(pont_arq, "\t");
226
227         };
228
229         if(j!=(quantidade_remedio[i][0]-1)){
230             fprintf(pont_arq, "\n");
231             fprintf(pont_arq, "\t\t\t");
232         };
233     };
234     fprintf(pont_arq, "\n");
235 };
236
237 //Fim do cadastro de pacientes
238 //
239 *****
240
241     inicializacao=1;
242     fclose(pont_arq);
243
244     return (0);
245 };
246
247 int ler_dados(){
248
249     //garantir variaveis zeradas
250     qtdp=0;
251     r_qtdp=-1;
252     i=0;

```

```

253 j=0;
254 a=0;
255
256
257
258 int cont=0;
259 int copia=0;
260 int mem[8];
261 int horario=0;
262 int laco=0;
263
264 pont_arq=fopen("teste3.txt","r");
265
266 while (!feof(pont_arq))
267 {
268 // Le uma linha (inclusive com o '\n')
269 result = fgets(leitura[r_qtdp], 400, pont_arq);
270 // o 'fgets' le ate 400 caracteres ou ate o '\n'
271
272 if (strlen(leitura[r_qtdp])<10 && inicializacao
273 ==1){ //se o arquivo reseteu o db ele resete
274 inicializacao tbm
275 inicializacao=0;
276 };
277
278 if (result){ // Se foi possivel ler
279
280 if (leitura[r_qtdp][0]!='\t' && laco>0){
281 qtdp++;
282 //printf("\nA quantidade de paciente eh : %d\n", qtdp);getchar();
283 copia=0;
284 cont=0;
285 mem[copia]=cont;
286
287 };

```

```

285 if (leitura[r_qtdp][0]!='\t' && leitura[r_qtdp]
286 ][1]!='\t'){
287 copia=3;
288 cont=3;
289 mem[copia]=cont;
290 quantidade_remedio[qtdp][0]=
291 quantidade_remedio[qtdp][0]+1;
292 strcpy(hora_lida1,"0");
293 strcpy(minuto_lida1,"0");
294 strcpy(hora_lida2,"0");
295 strcpy(minuto_lida2,"0");
296 strcpy(hora_lida4,"0");
297 strcpy(minuto_lida4,"0");
298
299 };
300
301 //printf("\nPaciente %d : %s", (qtdp),leitura[
302 r_qtdp]);
303
304 while (leitura[r_qtdp][cont]!='\n' && r_qtdp!= -1
305 && strlen(leitura[r_qtdp])>10){
306
307 if (leitura[r_qtdp][cont]!='\t'){
308 switch (copia){
309 case 0:
310 usuario[qtdp-1][cont]=leitura[r_qtdp][
311 cont];
312 //printf("\nO caracter usuario copiado
313 eh : %c\n",usuario[qtdp-1][cont]);getchar();
314 //printf("\nO cursor eh : %d\n", cont);
315 getchar();
316 break;
317
318 case 1:

```

```

315         idade_lida[cont-mem[copia]]=leitura[
r_qtdp][cont]; //cont - mem elimina a posicao
do \t e copia os caracteres
316         //printf("\nO caracter idade copiado eh
: %c\n", idade_lida[cont-mem[copia]]); getchar()
;
317         //printf("\nO cursor eh : %d\n", cont);
getchar();
318         break;
319
320     case 2:
321         if (atoi(idade_lida)!=0){
322             idade[qtdp-1]=atoi(idade_lida); //
grava a variavel
323             strcpy(idade_lida, "0"); //limpa a
variavel
324         };
325
326         remedio_lida[cont-mem[copia]]=leitura[
r_qtdp][cont];
327         //printf("\nO caracter QTD REMEDIO
copiado eh : %c\n", remedio_lida[cont-mem[copia]]); getchar();
328         //printf("\nO cursor eh : %d\n", cont);
getchar();
329         break;
330
331     case 3:
332         if (atoi(remedio_lida)!=0){
333             quantidade_remedio[qtdp-1][0]=atoi(
remedio_lida);
334             strcpy(remedio_lida, "0");
335         };
336
337
338         remedio[qtdp-1][quantidade_remedio[qtdp]
[0]][cont-mem[copia]]=leitura[r_qtdp][cont];
339

```

```

339         //printf("\nO caracterNOME REMEDIO
copiado eh : %c\n", remedio[qtdp-1][
quantidade_remedio[qtdp][0]][cont-mem[copia]]);
getchar();
340         //printf("\nO cursor eh : %d\n", cont);
getchar();
341         break;
342
343     case 4:
344         dose_lida[cont-mem[copia]]=leitura[
r_qtdp][cont];
345         //printf("\nO caracter qtd dose copiado
eh : %c\n", dose_lida[cont-mem[copia]]);
getchar();
346         //printf("\nO cursor eh : %d\n", cont);
getchar();
347         break;
348
349     case 5:
350         if (atoi(dose_lida)!=0){
351             quantidade_dose[qtdp-1][
quantidade_remedio[qtdp][0]][0]=atoi(dose_lida)
;
352             strcpy(dose_lida, "0");
353         };
354
355         if (leitura[r_qtdp][cont]!=':') {
356             if (horario==0){
357                 hora_lida1[cont-mem[copia]]=leitura
[r_qtdp][cont];
358                 //printf("\nO caracter hora1
copiado eh : %c\n", hora_lida1[cont-mem[copia]]); getchar();
359                 //printf("\nO cursor eh : %d\n",
cont); getchar(); printf("\nO cursor eh : %d\n",
cont); getchar();
360             } else {

```

```

361         minuto_lida1[cont-mem[copia]-3]=
leitura[r_qtdp][cont];
362         // printf("\nO caracter minuto1
copiado eh : %c\n", minuto_lida1[cont-mem[copia]
]-3]);getchar();
363         // printf("\nO cursor eh : %d\n",
cont);getchar();
364     }
365     }else{
366         horario=1;
367     };
368     quantidade_horario[qtdp-1][
quantidade_remedio[qtdp][0]][0]=1;
369     break;
370
371     case 6:
372         if( leitura[r_qtdp][cont]!=':'){
373             if(horario==0){
374                 hora_lida2[cont-mem[copia]]=leitura
[r_qtdp][cont];
375                 // printf("\nO caracter hora2
copiado eh : %c\n", hora_lida2[cont-mem[copia]
]);getchar();
376                 // printf("\nO cursor eh : %d\n",
cont);getchar();
377             }else{
378                 minuto_lida2[cont-mem[copia]-3]=
leitura[r_qtdp][cont];
379                 // printf("\nO caracter minuto2
copiado eh : %c\n", minuto_lida2[cont-mem[copia]
]-3]);getchar();
380                 // printf("\nO cursor eh : %d\n",
cont);getchar();
381             }
382         }else{
383             horario=1;
384         };

```

```

385         quantidade_horario[qtdp-1][
quantidade_remedio[qtdp][0]][0]=2;
386         break;
387
388         case 7:
389             if( leitura[r_qtdp][cont]!=':'){
390                 if(horario==0){
391                     hora_lida4[cont-mem[copia]]=leitura
[r_qtdp][cont];
392                     // printf("\nO caracter hora3
copiado eh : %c\n", hora_lida4[cont-mem[copia]
]);getchar();
393                     // printf("\nO cursor eh : %d\n",
cont);getchar();
394                 }else{
395                     minuto_lida4[cont-mem[copia]-3]=
leitura[r_qtdp][cont];
396                     // printf("\nO caracter minuto3
copiado eh : %c\n", minuto_lida4[cont-mem[copia]
]-3]);getchar();
397                     // printf("\nO cursor eh : %d\n",
cont);getchar();
398                 }
399             }else{
400                 horario=1;
401             };
402             quantidade_horario[qtdp-1][
quantidade_remedio[qtdp][0]][0]=3;
403             break;
404
405         }
406     }else{
407         copia++;
408         mem[copia]=cont+1;
409         horario=0;
410     }
411 }

```



```

412         cont++;
413     }
414 }
415 }
416 }
417 }
418
419     // *****
420     // conversao das horas para int
421     hora[qtdp-1][quantidade_remedio[qtdp][0]][0]=
atoi(hora_lida1);
422     minuto[qtdp-1][quantidade_remedio[qtdp
][0]][0]= atoi(minuto_lida1);
423     hora[qtdp-1][quantidade_remedio[qtdp][0]][1]=
atoi(hora_lida2);
424     minuto[qtdp-1][quantidade_remedio[qtdp
][0]][1]= atoi(minuto_lida2);
425     hora[qtdp-1][quantidade_remedio[qtdp][0]][2]=
atoi(hora_lida4);
426     minuto[qtdp-1][quantidade_remedio[qtdp
][0]][2]= atoi(minuto_lida4);
427     // *****
428
429     laco++;
430     r_qtdp++;
431 }
432 }
433
434     fclose(pont_arq);
435     return (0);
436
437 };
438
439 int mostrar_dados(){
440
441     printf("\n\nLeitura dos dados\n\n");
442

```

```

443     getchar();
444     int i=0;
445
446
447     //
448     *****
449
450     //Inicio do leitura do cadastro de pacientes
451
452     for (i=0;i<qtdp;i++){ //i = numero do paciente
453
454         //
455         *****
456
457         //ZERAR VARIAVEIS TIPO CONTADORES DAS LOGICAS
458         DE CADA USUARIO
459         j=0;
460         a=0;
461         //
462         *****
463
464         //Nome do usuario
465         printf("\nO nome do usuario %d e: %s", (i+1),
usuario[i]);
466         getchar();
467
468         //Idade do usuario
469         printf("\nA idade do usuario %d e: %d anos", (i
+1), idade[i]);
470         getchar();
471
472         //Quantidade de remedio
473         printf("\nA quantidade de remedios do usuario %
d e: %d ", (i+1), quantidade_remedio[i][0]);
474         getchar();
475

```

```

470     for(j=0;j<quantidade_remedio[i][0];j++){ //laco
471         do remedio
472
473         //Nome do remedio
474         printf("\nO nome do remedio %d do usuario %d
475 e: %s", (j+1), (i+1), remedio[i][j]);
476         getchar();
477
478         //Quantidade de dose do remedio
479         printf("\nA quantidade de dose do remedio %d
480 do usuario %d e: %d", (j+1), (i+1),
481 quantidade_dose[i][j][0]);
482         getchar();
483
484         //Quantidade de horarios do remedio
485         printf("\nA quantidade de horarios do remedio
486 %d do usuario %d e: %d", (j+1), (i+1),
487 quantidade_horario[i][j][0]);
488         getchar();
489
490         for(a=0;a<quantidade_horario[i][j][0];a++){
491             //Hora do remedio
492             printf("\nO horario %d do remedio %d do
493 usuario %d e: %d : %d", (a+1), (j+1), (i+1),
494 hora[i][j][a], minuto[i][j][a]);
495             getchar();
496
497         };
498
499     };
500
501 };
502
503 //Fim da leitura do cadastro de pacientes
504 //
505
506 *****
507
508     return (0);
509 };
510
511 int limpar_dados(){
512     pont_arq=fopen("teste3.txt","w");
513
514     //
515     *****
516
517     //prepara cabecalho do banco de dados
518     fprintf(pont_arq, "%s", "usuario");
519     fprintf(pont_arq, "\t");
520     fprintf(pont_arq, "%s", "idade");
521     fprintf(pont_arq, "\t");
522     fprintf(pont_arq, "%s", "Quantidade de remedios
523 ");
524     fprintf(pont_arq, "\t");
525     fprintf(pont_arq, "%s", "nome dos remedios");
526     fprintf(pont_arq, "\t");
527     fprintf(pont_arq, "%s", "Quantidade de dose");
528     fprintf(pont_arq, "\t");
529     fprintf(pont_arq, "%s", "horario 01");
530     fprintf(pont_arq, "\t");
531     fprintf(pont_arq, "%s", "horario 02");
532     fprintf(pont_arq, "\t");
533     fprintf(pont_arq, "%s", "horario 03");
534     fprintf(pont_arq, "\n");
535
536     //
537     *****

```

```
529     fclose(pont_arq);
530     return (0);
531 };
532
533
534 int chamar_codigo(int selecao){
535
536     switch (selecao){
537         case 1:
538             ler_dados();
539             break;
540         case 2:
541             escrever_dados();
542             break;
543         case 3:
544             mostrar_dados();
545             break;
546         case 4:
547             limpar_dados();
548             break;
549     };
550 };
551
552 return (0);
553 };
```