

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 da 1. 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]



Fig. 1. Sistema de gerenciamento de medicamentos já desenvolvido no mercado.

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, o 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*;
- O usuário responsável pelo sistema terá que inserir os comprimidos no dispositivo, assim que cada *slot* estiver desocupado.

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.

V. HARDWARE

Para o devido funcionamento do projeto, o dispositivo eletromecânico contará com os seguintes componentes: estilo

- 1) **Raspberry Pi:** A Raspberry Pi 3 é um computador de baixo custo e portátil, ela suporta o sistema operacional Ubuntu, Raspbian e outras distribuições

do Linux. Além disso, é compatível com o Windows 10 IoT, versão do software da Microsoft feita para automação doméstica e outras aplicações envolvendo Internet das Coisas.

O Raspberry Pi 3 pode ser encontrado no mercado com o valor em média de R\$ 200,00. Como na Fig. 2 a Raspberry Pi 3 suporta vários periféricos como quatro portas USB, uma porta HDMI, *WiFi* para conexão com a internet, *slot* para microSD e porta *ethernet*, aumentando suas aplicações com o uso da Raspberry Pi 3.



Fig. 2. Raspberry Pi 3

- 2) **Motor de passo:** Um motor de passo como na Fig.3 é um recurso eletromecânico onde precisa de um movimento controlado. A rotação do motor é definida com base nos pulsos elétricos gerados e a velocidade do motor de passo é definida pela frequência com que esses pulsos são enviados.

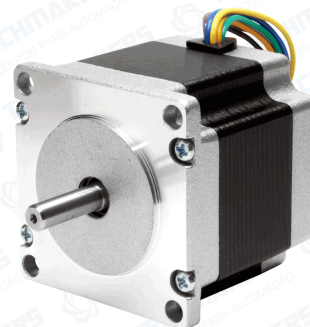


Fig. 3. Motor de passo.

VI. RESULTADOS

Foi desenvolvido um dispositivo eletromecânico a partir da impressão na impressora 3D, o desenvolvimento do desenho foi a partir do *software Fusion* da *Autodesk*. Pode-se observar nas figuras seguintes como será o dispositivo

para o gerenciamento de comprimidos:

Na Fig. 4 pode-se verificar como será a montagem, o mesmo terá diversos *slots* para inserir os comprimidos que suportará vários dias sem a necessidade de repor os comprimidos, os *slots* foi desenvolvido em forma circular para facilitar a rotação que será feita pelo motor de passo.

Também contará com uma base para coleta dos comprimidos que serão dispensados na hora que o usuário terá que ingeri-los, isso permitirá que o usuário sempre ingira os comprimidos adequados para seu tratamento na hora certa.

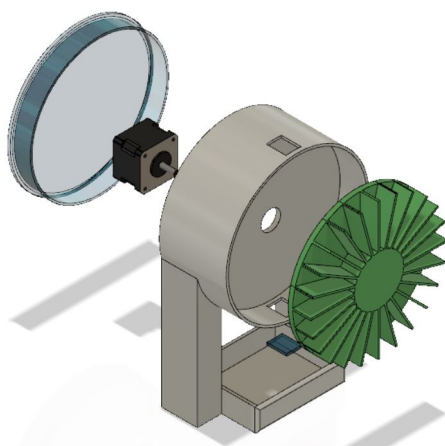


Fig. 4. Vista explodida do dispositivo

Na Fig. 5 e na Fig. 6 pode-se verificar o produto final:

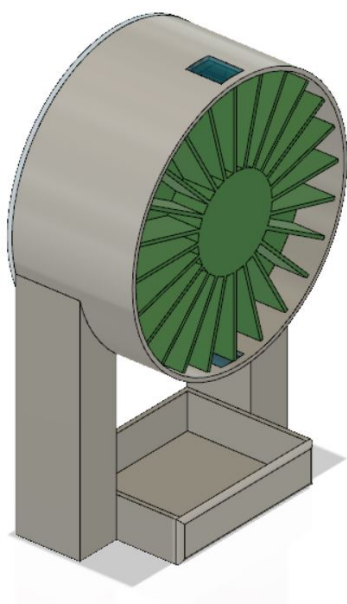


Fig. 5. Dispositivo eletromecânico para dispensar os comprimidos.

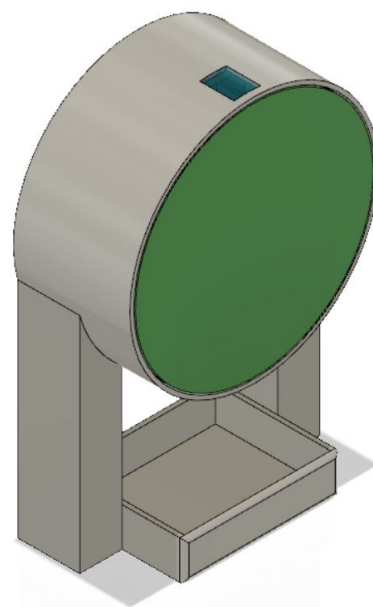


Fig. 6. Produto final.

VII. CONSIDERAÇÕES FINAIS

Em vista do exposto, o projeto está de acordo com o que foi proposto para a matéria de sistemas embarcados, foi possível até este ponto de controle permitir ao administrador do sistema configurar a rotina de horários e quantidade de remédios a serem prescritos, além de cadastrar os usuários por meio de um banco de dados e apresentar um dispositivo eletromecânico para dispensar o remédio automaticamente por meio da Raspberry pi. Para o próximo ponto de controle será possível ter a interface gráfica facilitando a interação com o usuário, assim como o processamento de imagens para identificação do usuário.

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. Kourrouski 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. Siudzinski, "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>.

```

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     //abre arquivo
48     pont_arq=fopen("teste3.txt","r");
49
50     if (pont_arq == NULL ) {
51         printf("\n**** arquivo nao existe ****\n");
52         pont_arq=fopen("teste3.txt","w");
53
54         //cabecalho
55         fprintf(pont_arq, "%s", "usuario");
56         fprintf(pont_arq, "\t");
57         fprintf(pont_arq, "%s", "idade");
58         fprintf(pont_arq, "\t");
59         fprintf(pont_arq, "%s", "Quantidade de remedios");
60         fprintf(pont_arq, "\t");
61         fprintf(pont_arq, "%s", "nome dos remedios");
62         fprintf(pont_arq, "\t");
63         fprintf(pont_arq, "%s", "Quantidade de dose");
64         fprintf(pont_arq, "\t");
65         fprintf(pont_arq, "%s", "horario 01");
66         fprintf(pont_arq, "\t");
67         fprintf(pont_arq, "%s", "horario 02");
68         fprintf(pont_arq, "\t");
69         fprintf(pont_arq, "%s", "horario 03");
70         fprintf(pont_arq, "\n");
71
72     } else {
73

```

```

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     //
143     *****

```

142		168	fprintf(pont_arq , "%s" , usuario[i]);
143		169	fprintf(pont_arq , "\t");
144		170	
145	//	171	//Idade do usuario
	*****	172	if(i==mem inicializacao<1){ //garante a nao
		173	escrita sobre o dado anterior
146	//Inicio do cadastro de pacientes		printf("\nEntre com a idade do usuario %d: ",
147		174	(i+1));
148	printf("\nEntre com a quantidade de pacientes: ")	175	scanf("%d" , &idade[i]);getchar();
149	;	176	};
150	scanf("%d" , &qtd);	177	fprintf(pont_arq , "%d" , idade[i]);
151	getchar();	178	fprintf(pont_arq , "\t");
152	mem=qtdp;	179	
153	qtdp=qtd+qtdp;	180	//Quantidade de remedio
154		181	if(i==mem inicializacao<1){ //garante a nao
155	for (i=0;i<qtdp;i++){ //i = numero do paciente	182	escrita sobre o dado anterior
156		183	printf("\nEntre com a quantidade de remedios
157	//	184	do usuario %d: ", (i+1));
	*****	185	scanf("%d" , &quantidade_remedio[i][0]);
		186	getchar();
158	//ZERAR VARIAVEIS TIPO CONTADORES DAS LOGICAS	187	};
159	DE CADA USUARIO	188	fprintf(pont_arq , "%d" , quantidade_remedio[i
160	j=0;	189][0]);
161	a=0;	190	fprintf(pont_arq , "\t");
	//	191	
	*****	192	for(j=0;j<quantidade_remedio[i][0];j++){ //laco
		193	do remedio
162		194	
163	//Nome do usuario	195	//Nome do remedio
164	if(i==mem inicializacao<1){ //garante a nao		if(i==mem inicializacao<1){ //garante a
	escrita sobre o dado anterior		nao escrita sobre o dado anterior
165	printf("\nEntre com o nome do usuario %d: ",		printf("\nEntre com o nome do remedio %d do
	(i+1));		usuario %d: ", (j+1), (i+1));
166	gets(usuario[i]);		gets(remedio[i][j]);
167	};		};
			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     if(j!=(quantidade_remedio[i][0]-1)){
229         fprintf(pont_arq, "\n");
230         fprintf(pont_arq, "\t\t\t");
231     };
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 resetou 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 };
288 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:
319 idade_lida[cont-mem[copia]]=leitura[

```



```

316 r_qtdp][cont]; //cont - mem elimina a posicao
do \t e copia os caracteres
317 //printf("\nO caracter idade copiado eh
: %c\n", idade_lida[cont-mem[copia]]); getchar() 340
;
318 //printf("\nO cursor eh : %d\n", cont); 341
getchar(); 342
319 break; 343
320 case 2: 344
321 if (atoi(idade_lida) != 0){
322 idade[qtdp-1] = atoi(idade_lida); //
grava a variavel
323 strcpy(idade_lida, "0"); //limpa a
variavel 346
} 347
324 348
325 remedio_lida[cont-mem[copia]] = leitura[
r_qtdp][cont]; 349
326 //printf("\nO caracter QTD REMEDIO
copiado eh : %c\n", remedio_lida[cont-mem[copia]]); getchar(); 350
327 //printf("\nO cursor eh : %d\n", cont); 351
getchar(); 352
328 break; 353
329 354
330 case 3: 355
331 if (atoi(remedio_lida) != 0){
332 quantidade_remedio[qtdp-1][0] = atoi(
remedio_lida); 356
333 strcpy(remedio_lida, "0"); 357
} 358
334 359
335 remedio[qtdp-1][quantidade_remedio[qtdp
][0]][cont-mem[copia]] = leitura[r_qtdp][cont]; 360
336 //printf("\nO caracter NOME REMEDIO 361

```

```

copiado eh : %c\n", remedio[qtdp-1][
quantidade_remedio[qtdp][0]][cont-mem[copia]]);
getchar();
//printf("\nO cursor eh : %d\n", cont);
getchar();
break;

case 4:
dose_lida[cont-mem[copia]] = leitura[
r_qtdp][cont];
//printf("\nO caracter qtd dose copiado
eh : %c\n", dose_lida[cont-mem[copia]]);
getchar();
//printf("\nO cursor eh : %d\n", cont);
getchar();
break;

case 5:
if (atoi(dose_lida) != 0){
quantidade_dose[qtdp-1][
quantidade_remedio[qtdp][0]][0] = atoi(dose_lida)
;
strcpy(dose_lida, "0");
};

if (leitura[r_qtdp][cont] != ':'){
if (horario == 0){
hora_lida1[cont-mem[copia]] = leitura
[r_qtdp][cont];
//printf("\nO caracter hora1
copiado eh : %c\n", hora_lida1[cont-mem[copia]]);
getchar();
//printf("\nO cursor eh : %d\n",
cont); getchar(); printf("\nO cursor eh : %d\n",
cont); getchar();
} else {
minuto_lida1[cont-mem[copia]-3] =

```

```

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][

```

```

386     quantidade_remedio [ qtdp ][ 0 ][ 0 ] = 2;
387     break;
388
389     case 7:
390         if ( leitura [ r_qtdp ][ cont ] != ':' ) {
391             if ( horario == 0 ) {
392                 hora_lida4 [ cont - mem [ copia ] ] = leitura
[ r_qtdp ][ cont ];
393                 // printf ( "\nO caracter hora3
copiado eh : %c\n", hora_lida4 [ cont - mem [ copia
394                 ] ] ); getchar ();
395                 // printf ( "\nO cursor eh : %d\n",
cont ); getchar ();
396             } else {
397                 minuto_lida4 [ cont - mem [ copia ] - 3 ] =
leitura [ r_qtdp ][ cont ];
398                 // printf ( "\nO caracter minuto3
copiado eh : %c\n", minuto_lida4 [ cont - mem [ copia
399                 ] - 3 ] ); getchar ();
400                 // printf ( "\nO cursor eh : %d\n",
cont ); getchar ();
401             }
402         } else {
403             horario = 1;
404
405             quantidade_horario [ qtdp - 1 ][
406             quantidade_remedio [ qtdp ][ 0 ][ 0 ] = 3;
407             break;
408
409         }
410     } else {
411         copia ++;
412         mem [ copia ] = cont + 1;
413         horario = 0;
414     }
415 }
416

```

```

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

```

```

444 int i=0;
445
446 //
447 // *****
448 //Inicio do leitura do cadastro de pacientes
449
450 for (i=0;i<qtdp;i++){ //i = numero do paciente
451
452     //
453     // *****
454
455     //ZERAR VARIABEIS TIPO CONTADORES DAS LOGICAS
456     DE CADA USUARIO
457     j=0;
458     a=0;
459     //
460     // *****
461
462     //Nome do usuario
463     printf("\nO nome do usuario %d e: %s", (i+1),
464     usuario[i]);
465     getchar();
466
467     //Idade do usuario
468     printf("\nA idade do usuario %d e: %d anos", (i
469     +1), idade[i]);
470     getchar();
471
472     //Quantidade de remedio
473     printf("\nA quantidade de remedios do usuario %
474     d e: %d ", (i+1), quantidade_remedio[i][0]);
475     getchar();
476
477     for (j=0;j<quantidade_remedio[i][0];j++){ //laco

```

```

do remedio
//Nome do remedio
printf("\nO nome do remedio %d do usuario %d
e: %s", (j+1), (i+1), remedio[i][j]);
getchar();

//Quantidade de dose do remedio
printf("\nA quantidade de dose do remedio %d
do usuario %d e: %d", (j+1), (i+1),
quantidade_dose[i][j][0]);
getchar();

//Quantidade de horarios do remedio
printf("\nA quantidade de horarios do remedio
%d do usuario %d e: %d", (j+1), (i+1),
quantidade_horario[i][j][0]);
getchar();

for(a=0;a<quantidade_horario[i][j][0];a++){
//Hora do remedio
printf("\nO horario %d do remedio %d do
usuario %d e: %d : %d", (a+1), (j+1), (i+1),
hora[i][j][a], minuto[i][j][a]);
getchar();
};

};

};

//Fim da leitura do cadastro de pacientes

```

```

//
*****

return (0);
};

int limpar_dados(){
pont_arq=fopen("teste3.txt","w");

//
*****

//prepara cabecalho do banco de dados
fprintf(pont_arq, "%s", "usuario");
fprintf(pont_arq, "\t");
fprintf(pont_arq, "%s", "idade");
fprintf(pont_arq, "\t");
fprintf(pont_arq, "%s", "Quantidade de remedios
");
fprintf(pont_arq, "\t");
fprintf(pont_arq, "%s", "nome dos remedios");
fprintf(pont_arq, "\t");
fprintf(pont_arq, "%s", "Quantidade de dose");
fprintf(pont_arq, "\t");
fprintf(pont_arq, "%s", "horario 01");
fprintf(pont_arq, "\t");
fprintf(pont_arq, "%s", "horario 02");
fprintf(pont_arq, "\t");
fprintf(pont_arq, "%s", "horario 03");
fprintf(pont_arq, "\n");

//
*****

fclose(pont_arq);

```

```

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

```

APÊNDICE - Implementação da rotação com o motor de passo

```

1  #include "gpio-sysfs.h"
2  #include <unistd.h>
3
4  int main()
5  {
6      int in1=4, in2=14, in3=17, in4=18;
7
8      //Configurando os pinos como saia
9      setGPIO_Out(in1);

```

```

10     setGPIO_Out(in2);
11     setGPIO_Out(in3);
12     setGPIO_Out(in4);
13
14     while(1)
15     {
16         GPIO_Write(in1,1);
17         GPIO_Write(in2,0);
18         GPIO_Write(in3,0);
19         GPIO_Write(in4,1);
20         usleep(10000);
21
22         GPIO_Write(in1,0);
23         GPIO_Write(in2,1);
24         GPIO_Write(in3,0);
25         GPIO_Write(in4,1);
26         usleep(10000);
27
28         GPIO_Write(in1,0);
29         GPIO_Write(in2,1);
30         GPIO_Write(in3,1);
31         GPIO_Write(in4,0);
32         usleep(10000);
33
34         GPIO_Write(in1,1);
35         GPIO_Write(in2,0);
36         GPIO_Write(in3,1);
37         GPIO_Write(in4,0);
38         usleep(10000);
39     }
40
41 }

```