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 vêem-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.
- 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:

Tabela 1: Materiais utilizados na confecção do protótipo

Materiais	Quantidade (und)
Raspberry Pi 3	1
Servo motor	2
Motor de passo	1
Led's	3
Estrutura feita na impressora 3	1
Webcam	1
Monitor	1
Buzzer	1

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 e servomotor**: Um motor de passo como na Fig.3 e Fig. 4 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. O motor e o servo motor servirão para o acionamento da movimentação dos *slots*.

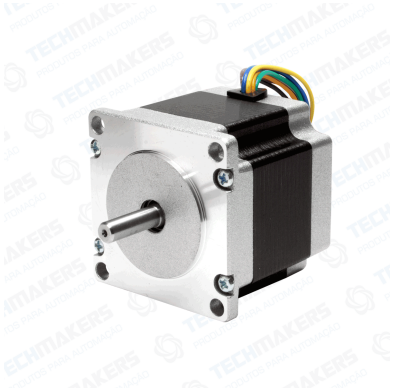

Fig. 3. Motor de passo NEMA.

3) **Buzzer**:
Buzzer é um componente eletrônico da Fig. 5 que recebe uma fonte de energia e através dela emite uma

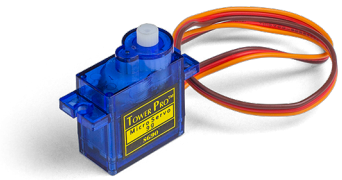

Fig. 4. Servo motor.

frequência sonora, com isso avisará aos usuários do sistema que está na hora de ingerir o remédio.

Fig. 5. Buzzer.

A conexão destes elementos com a *Raspberry Pi* foi feita utilizando os pinos de GPIO e fazendo uso de *jumpers*. A numeração destes pinos obedeceu a estipulada na biblioteca *WiringPi* e pode ser verificada nos códigos em anexo a este documento. Pode-se visualizar estas conexões na Fig. 6 abaixo. Junto a estes componentes, há uma webcam conectada na porta USB da placa.

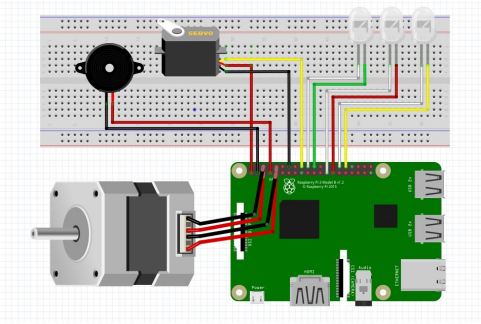

Fig. 6. Conexões componentes periféricos do sistema.

VI. *SOFTWARE*

Os programas utilizados para movimentação do equipamento estão disponíveis no apêndice deste documento. O funcionamento será da seguinte forma: O usuário deverá utilizar o programa de cadastramento previamente para gravar as informações do paciente no banco de dados. Após isso, será gerado um arquivo para armazenar as informações.

Com o banco de dados Fig. 7 preenchido, o programa que estipula os alarmes dos remédio também é o responsável por recolher as informações no banco de dados e de chamar os códigos dos periféricos necessários para dispensar o medicamento no momento correto. Por exemplo, ao identificar um horário de medicação, o programa lê as informações no banco de dados com o nome e o *slot* onde está localizado o comprimido.

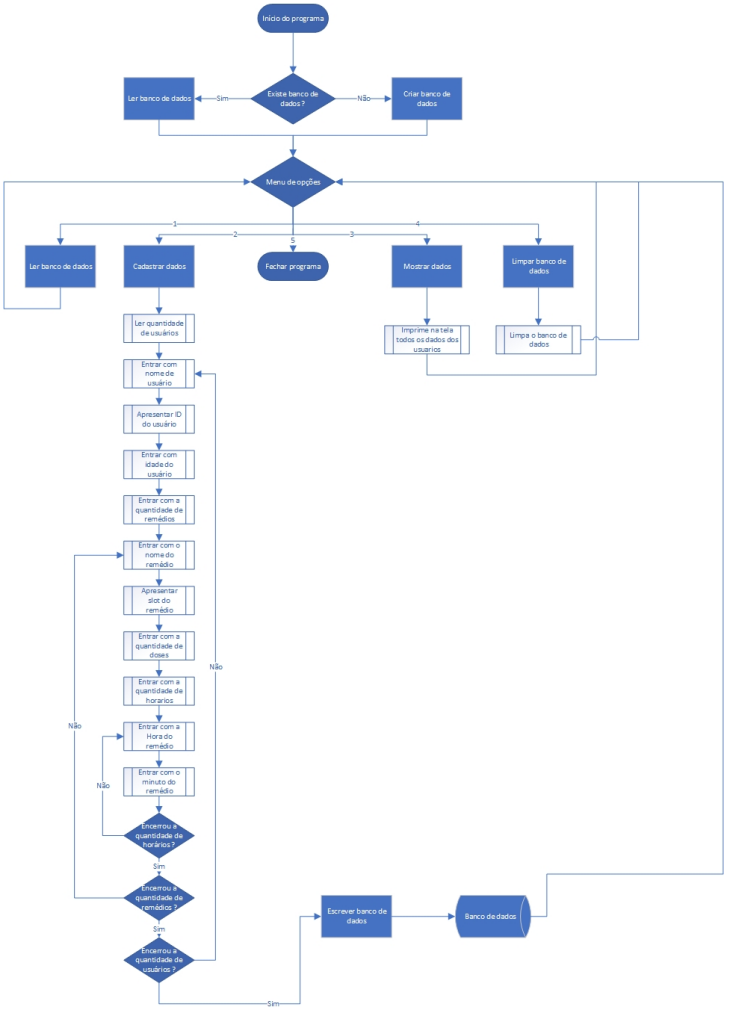

Fig. 7. Fluxograma do banco de dados.

Em seguida, aciona o motor de passo para o giro do eixo principal do programa de forma a posicionar o comprimido em cima da comporta. Após isso é emitido o aviso sonoro Fig. 8 ao usuário indicando a necessidade de se aproximar

do aparelho. Quando o usuário se aproxima do dispositivo, sua imagem é capturada pela *Webcam* acoplada ao sistema e a partir deste ponto é realizado um processamento de imagem para identificar se a pessoa certa receberá os comprimidos. Uma vez identificada, cessam os avisos sonoros e o servo motor é acionado liberando os dispositivos.

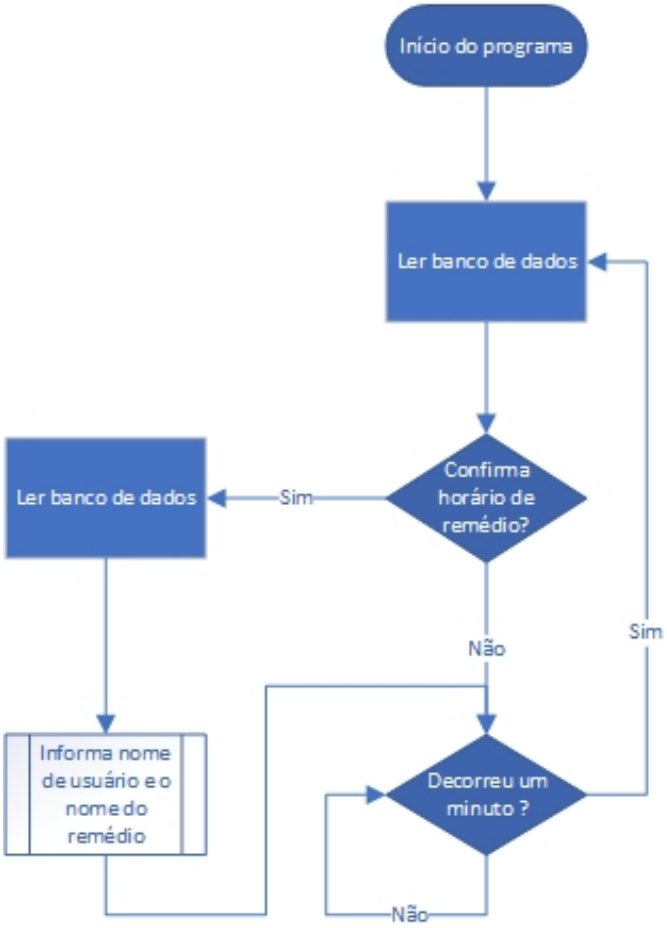

Fig. 8. Fluxograma da hora do acionamento do BUzzer.

Há também algumas indicações luminosas para facilitar o entendimento do processo. Ao iniciar o aviso sonoro uma luz amarela piscará de forma intermitente. Ao iniciar a identificação visual do usuário a luz manterá o brilho constante. Caso seja liberado o comprimido para o usuário, um LED verde acenderá. Caso contrário, um LED vermelho irá acender.

O acionamento dos componentes periféricos é feito basicamente utilizando a função *system()* e passando os parâmetros necessários. Entretanto, para algumas funções que necessitam funcionar paralelamente a outras será necessário criar processos filhos ou *threads*.

VII. 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. 9 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.

Os resultados parciais do desenvolvimento deste projeto promovem uma confiança na finalização satisfatória deste dispositivo. Pode-se desenvolver todos os códigos necessários para trabalhar isoladamente com cada componente da estrutura, restando desta forma apenas integrá-los para o funcionamento sincronizado. Algumas pequenas mudanças serão necessárias para garantir a maior eficiência no uso dos recursos do sistema adicionando o uso de *threads*, por exemplo.

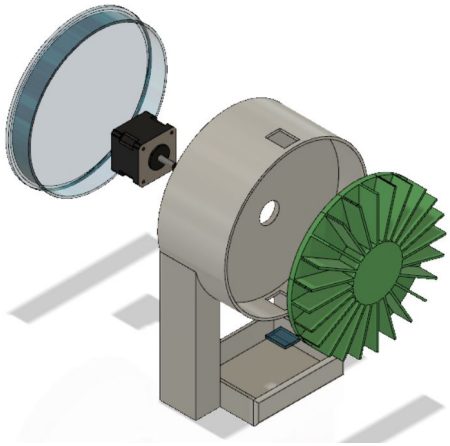

Fig. 9. Vista explodida do dispositivo

Na Fig. 10 e na Fig. 11 pode-se verificar o produto final:

VIII. 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.

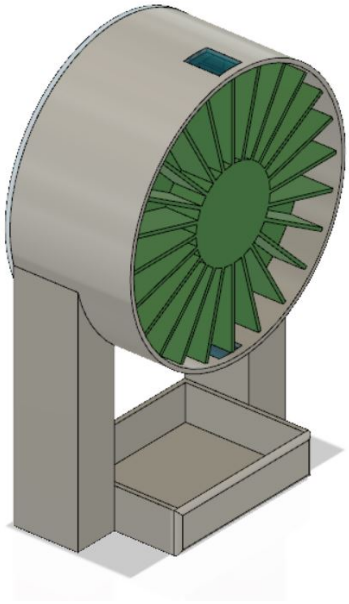

Fig. 10. Dispositivo eletromecânico para dispensar os comprimidos.

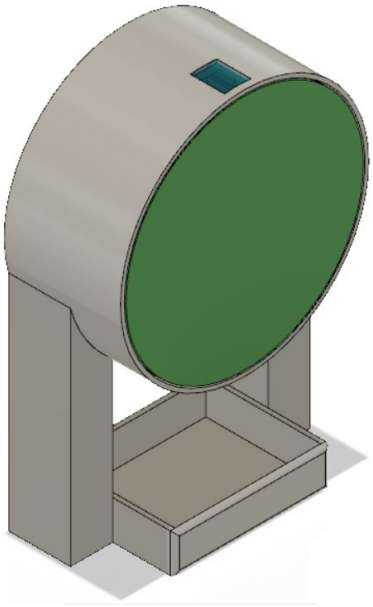

Fig. 11. Produto final.

Foi possível aplicar os conhecimentos adquiridos durante a disciplina. Para o banco de dados, por exemplo, foi utilizado o conhecimento de arquivos visto no início do curso. Além disso, a comunicação entre os diversos componentes é feita pela troca de informações através de parâmetros no início da chamada do sistema onde é necessário a teoria da programação em C com ênfase em ponteiros. Adicionalmente, é possível observar a teoria de sistemas embarcados nas lógicas de programação empregadas em especial na criação de processos simultâneos.

Para a entrega do relatório final, será feito a integração do sistema de forma a garantir o funcionamento sincronizado dos diversos componentes além de um refinamento nos códigos de programação e o acabamento final da estrutura.

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 pa- ciente 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. Siudzinzki, "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.

APÊNDICE - Implementação do Banco de Dados

```
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 mostar_dados();
39 int limpar_dados();
40 int chamar_codigo(int selecao);
41
42 //Criar ponteiro de arquivo
43 FILE *pont_arq;
44
```

```c
int main(){


  //abre arquivo
  pont_arq=fopen("teste3.txt","r");

  if (pont_arq == NULL ) {
    printf("\n****  arquivo nao existe ****\n");
    pont_arq=fopen("teste3.txt","w");

    //cabecalho
    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");

  }else{
    printf("\n****  arquivo aberto com sucesso
    ****\n");
    ler_dados();
    inicializacao=1;
  };

  fclose(pont_arq);


  while (selecao!=5){
    while (selecao<1 || selecao>5){
      printf("\n****  Qual modo de operacao deseja
    ****\n");
      printf("\n****  Ler banco de dados : 1 ****\n
    ");
      printf("\n****  cadastrar dados : 2 ****\n");
      printf("\n****  Mostrar dados : 3 ****\n");
      printf("\n****  Limpar banco de dados : 4
    ****\n");
      printf("\n****  Fechar programa : 5 ****\n");
      scanf("%d", &selecao);
    };

    printf("\nModo selecionado : %d\n", selecao);

    chamar_codigo(selecao);

    fclose(pont_arq);

    if (selecao<5 && selecao!=0){
      selecao=0;
    };


  };


  fclose(pont_arq);
  printf("\nARQUIVO FECHADO\n");
  getchar();

  return (0);
}
```

```
int escrever_dados(){

int qtd=0;
int mem;

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");

//
    ************************************************



//
    ************************************************
```

```
//Inicio do cadastro de pacientes

  printf("\nEntre com a quantidade de pacientes: ")
    ;
  scanf("%d", &qtd);
  getchar();

  mem=qtdp;
  qtdp=qtd+qtdp;

  for (i=0;i<qtdp;i++){ //i = numero do paciente

    //
    ************************************************

    //ZERAR VARIAVEIS TIPO CONTADORES DAS LOGICAS
    DE CADA USUARIO
    j=0;
    a=0;
    //
    ************************************************


    //Nome do usuario
    if(i==mem || inicializacao<1){  //garante a nao
     escrita sobre o dado anterior
      printf("\nEntre com o nome do usuario %d: ",
    (i+1));
      gets(usuario[i]);
    };
    fprintf(pont_arq, "%s", usuario[i]);
    fprintf(pont_arq, "\t");

    //Idade do usuario
    if(i==mem || inicializacao<1){  //garante a nao
     escrita sobre o dado anterior
      printf("\nEntre com a idade do usuario %d: ",
```

```
 (i+1));
  scanf("%d", &idade[i]);getchar();
};
fprintf(pont_arq, "%d", idade[i]);
fprintf(pont_arq, "\t");

//Quantidade de remedio
if(i==mem || inicializacao<1){  //garante a nao escrita sobre o dado anterior
  printf("\nEntre com a quantidade de remedios do usuario %d: ", (i+1));
  scanf("%d", &quantidade_remedio[i][0]);getchar();
};
fprintf(pont_arq, "%d", quantidade_remedio[i][0]);
fprintf(pont_arq, "\t");


for(j=0;j<quantidade_remedio[i][0];j++){ //laco do remedio

  //Nome do remedio
  if(i==mem || inicializacao<1){  //garante a nao escrita sobre o dado anterior
    printf("\nEntre com o nome do remedio %d do usuario %d: ", (j+1), (i+1));
    gets(remedio[i][j]);
  };
  fprintf(pont_arq, "%s", remedio[i][j]);
  fprintf(pont_arq, "\t");

  //Quantidade de dose do remedio
  if(i==mem || inicializacao<1){  //garante a nao escrita sobre o dado anterior
    printf("\nEntre com a quantidade de dose do remedio %d do usuario %d: ", (j+1), (i+1));
    scanf("%d", &quantidade_dose[i][j][0]);getchar();
  };
  fprintf(pont_arq, "%d", quantidade_dose[i][j][0]);
  fprintf(pont_arq, "\t");

  //Quantidade de horarios do remedio
  if(i==mem || inicializacao<1){  //garante a nao escrita sobre o dado anterior
    printf("\nEntre com a quantidade de horarios do remedio %d do usuario %d: ", (j+1), (i+1));
    scanf("%d", &quantidade_horario[i][j][0]);getchar();
  };

  for(a=0;a<quantidade_horario[i][j][0];a++){
    //Hora do remedio
    if(i==mem || inicializacao<1){  //garante a nao escrita sobre o dado anterior
      printf("\nEntre com a hora %d do remedio %d do usuario %d: ", (a+1), (j+1), (i+1));
      scanf("%d", &hora[i][j][a]);getchar();
    };

    //Minuto do remedio
    if(i==mem || inicializacao<1){  //garante a nao escrita sobre o dado anterior
      printf("\nEntre com o minuto %d do remedio %d do usuario %d: ", (a+1), (j+1), (i+1));
      scanf("%d", &minuto[i][j][a]);getchar();
    };
    fprintf(pont_arq, "%d:%d", hora[i][j][a], minuto[i][j][a]);
    fprintf(pont_arq, "\t");
```

```
      };

      if(j!=(quantidade_remedio[i][0]-1)){
        fprintf(pont_arq, "\n");
        fprintf(pont_arq, "\t\t\t");
      };

    };
    fprintf(pont_arq, "\n");
  };

//Fim do cadastro de pacientes
//
    ***********************************************

  inicializacao=1;
  fclose(pont_arq);

  return (0);
};

int ler_dados(){

//garantir variaveis zeradas
  qtdp=0;
  r_qtdp=-1;
  i=0;
  j=0;
  a=0;



int cont=0;
int copia=0;
int mem[8];
int horario=0;
int laco=0;

pont_arq=fopen("teste3.txt","r");

  while (!feof(pont_arq))
  {
  // Le uma linha (inclusive com o '\n')
  result = fgets(leitura[r_qtdp], 400, pont_arq);
    // o 'fgets' le ate 400 caracteres ou ate o '\n'
  if(strlen(leitura[r_qtdp])<10 && inicializacao==1){ //se o arquivo resetou o db ele resete inicializacao tbm
    inicializacao=0;
  };

  if (result){  // Se foi possivel ler


    if(leitura[r_qtdp][0]!='\t' && laco>0){
      qtdp++;
      //printf("\nA quantidade de paciente eh : %d\n", qtdp);getchar();
      copia=0;
      cont=0;
      mem[copia]=cont;

    };
    if(leitura[r_qtdp][0]=='\t' && leitura[r_qtdp][1]=='\t'){
      copia=3;
      cont=3;
      mem[copia]=cont;
      quantidade_remedio[qtdp][0]=quantidade_remedio[qtdp][0]+1;
      strcpy(hora_lida1,"0");
```

```
    strcpy(minuto_lida1,"0");
    strcpy(hora_lida2,"0");
    strcpy(minuto_lida2,"0");
    strcpy(hora_lida4,"0");
    strcpy(minuto_lida4,"0");

};


//printf("\nPaciente %d : %s", (qtdp),leitura[r_qtdp]);


while(leitura[r_qtdp][cont]!='\n' && r_qtdp!=-1 && strlen(leitura[r_qtdp])>10){


  if(leitura[r_qtdp][cont]!='\t'){
    switch (copia){
      case 0:
        usuario[qtdp-1][cont]=leitura[r_qtdp][cont];
        //printf("\nO caracter usuario copiado eh : %c\n",usuario[qtdp-1][cont]);getchar();
        //printf("\nO cursor eh : %d\n", cont);getchar();
      break;

      case 1:
        idade_lida[cont-mem[copia]]=leitura[r_qtdp][cont];  //cont - mem elimina a posicao do \t e copia os caracteres
        //printf("\nO caracter idade copiado eh : %c\n",idade_lida[cont-mem[copia]]);getchar();
        //printf("\nO cursor eh : %d\n", cont);getchar();
      break;

      case 2:
        if(atoi(idade_lida)!=0){
          idade[qtdp-1]=atoi(idade_lida); //grava a variavel
          strcpy(idade_lida,"0");    //limpa a variavel
        };

        remedio_lida[cont-mem[copia]]=leitura[r_qtdp][cont];
        //printf("\nO caracter QTD REMEDIO copiado eh : %c\n", remedio_lida[cont-mem[copia]]);getchar();
        //printf("\nO cursor eh : %d\n", cont);getchar();
      break;

      case 3:
        if(atoi(remedio_lida)!=0){
          quantidade_remedio[qtdp-1][0]=atoi(remedio_lida);
          strcpy(remedio_lida,"0");
        };


        remedio[qtdp-1][quantidade_remedio[qtdp][0]][cont-mem[copia]]=leitura[r_qtdp][cont];
        //printf("\nO caracterNOME REMEDIO 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]=leitura[r_qtdp][cont];
              //printf("\nO caracter minuto1 copiado eh : %c\n", minuto_lida1[cont-mem[copia]-3]);getchar();
              //printf("\nO cursor eh : %d\n", cont);getchar();
            }
          }else{
            horario=1;
          };
          quantidade_horario[qtdp-1][quantidade_remedio[qtdp][0]][0]=1;
        break;

        case 6:
          if(leitura[r_qtdp][cont]!=':'){
            if(horario==0){
              hora_lida2[cont-mem[copia]]=leitura[r_qtdp][cont];
              //printf("\nO caracter hora2 copiado eh : %c\n", hora_lida2[cont-mem[copia]]);getchar();
              //printf("\nO cursor eh : %d\n", cont);getchar();
            }else{
              minuto_lida2[cont-mem[copia]-3]=leitura[r_qtdp][cont];
              //printf("\nO caracter minuto2 copiado eh : %c\n", minuto_lida2[cont-mem[copia]-3]);getchar();
              //printf("\nO cursor eh : %d\n", cont);getchar();
            }
          }else{
            horario=1;
          };
          quantidade_horario[qtdp-1][quantidade_remedio[qtdp][0]][0]=2;
        break;

        case 7:
          if(leitura[r_qtdp][cont]!=':'){
            if(horario==0){
              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         }
407       }else{
408         copia++;
409         mem[copia]=cont+1;
410         horario=0;
411       }
412
413       cont++;
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
440 int mostar_dados(){
441
442 printf("\n\nLeitura dos dados\n\n");
443 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     //ZERAR VARIAVEIS TIPO CONTADORES DAS LOGICAS
      DE CADA USUARIO
454     j=0;
455     a=0;
456     //
      **********************************************
457
458     //Nome do usuario
459     printf("\nO nome do usuario %d e: %s", (i+1),
      usuario[i]);
460     getchar();
461
462     //Idade do usuario
463     printf("\nA idade do usuario %d e: %d anos", (i
      +1), idade[i]);
464     getchar();
465
466     //Quantidade de remedio
467     printf("\nA quantidade de remedios do usuario %
      d e: %d ", (i+1), quantidade_remedio[i][0]);
468     getchar();
469
470     for(j=0;j<quantidade_remedio[i][0];j++){ //laco
       do remedio
471
472       //Nome do remedio
473       printf("\nO nome do remedio %d do usuario %d
      e: %s", (j+1), (i+1), remedio[i][j]);
474       getchar();
475
476       //Quantidade de dose do remedio
477       printf("\nA quantidade de dose do remedio %d
      do usuario %d e: %d", (j+1), (i+1),
      quantidade_dose[i][j][0]);
          getchar();
479
480       //Quantidade de horarios do remedio
481       printf("\nA quantidade de horarios do remedio
       %d do usuario %d e: %d", (j+1), (i+1),
      quantidade_horario[i][j][0]);
          getchar();
482
484       for(a=0;a<quantidade_horario[i][j][0];a++){
485         //Hora do remedio
486         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]);
487         getchar();
488
489       };
490
491
492
493     };
494
495
496
497   };
498
499 //Fim da leitura do cadastro de pacientes
500 //
      **********************************************
501
502
503   return (0);
504 };
```

```
505
506 int limpar_dados(){
507 pont_arq=fopen("teste3.txt","w");
508
509 //
    ***********************************************
510 //prepara cabecalho do banco de dados
511     fprintf(pont_arq, "%s", "usuario");
512     fprintf(pont_arq, "\t");
513     fprintf(pont_arq, "%s", "idade");
514     fprintf(pont_arq, "\t");
515     fprintf(pont_arq, "%s", "Quantidade de remedios
    ");
516     fprintf(pont_arq, "\t");
517     fprintf(pont_arq, "%s", "nome dos remedios");
518     fprintf(pont_arq, "\t");
519     fprintf(pont_arq, "%s", "Quantidade de dose");
520     fprintf(pont_arq, "\t");
521     fprintf(pont_arq, "%s", "horario 01");
522     fprintf(pont_arq, "\t");
523     fprintf(pont_arq, "%s", "horario 02");
524     fprintf(pont_arq, "\t");
525     fprintf(pont_arq, "%s", "horario 03");
526     fprintf(pont_arq, "\n");
527 //
    ***********************************************
528
529   fclose(pont_arq);
530   return (0);
531 };
532
533
534 int chamar_codigo(int selecao){
535
536
```

```
537   switch (selecao){
538     case 1:
539       ler_dados();
540     break;
541     case 2:
542       escrever_dados();
543     break;
544     case 3:
545       mostar_dados();
546     break;
547     case 4:
548       limpar_dados();
549     break;
550   };
551
552   return (0);
553 };
```

APÊNDICE - Implementação do código para acionamento do motor de passo

```
1  #include <stdio.h>
2  #include <wiringPi.h>
3  #include <stdlib.h>
4
5  int  in1 = 7;
6  int  in2 = 15;
7  int  in3 = 0;
8  int  in4 = 1;
9
10 const int calPasso = 18;
11 const int del = 5;
12
13 int passo = 0;
14
15 int movimenta(int passos, int direcao){
16
```

```
  if (direcao == 1 )
  {
    for (int i=0; i<passos; i++)
    {
      switch(passo)
      {
      case 4:
      case 0:
                digitalWrite(in1, HIGH);
                           digitalWrite(in2, LOW);
                           digitalWrite(in3, LOW);
                           digitalWrite(in4, HIGH);
                           delay(del);
              passo = 1;
        break;
            case 1:
        digitalWrite(in1, LOW);
                        digitalWrite(in2, HIGH);
                        digitalWrite(in3, LOW);
                            digitalWrite(in4, HIGH);
                      delay(del);
        passo = 2;
        break;
      case 2:
        digitalWrite(in1, LOW);
                        digitalWrite(in2, HIGH);
                        digitalWrite(in3, HIGH);
                            digitalWrite(in4, LOW);
                      delay(del);
        passo = 3;
              break;
            case 3:
                           digitalWrite(in1, HIGH);
                        digitalWrite(in2, LOW);
                        digitalWrite(in3, HIGH);
                            digitalWrite(in4, LOW);
                      delay(del);
        passo = 4;
        break;
      }
    }
  }
  else if (direcao == 2)
  {
    for (int i=0; i<passos; i++)
    {
      switch(passo)
      {
      case 0:
      case 1:
                  digitalWrite(in1, HIGH);
                      digitalWrite(in2, LOW);
                      digitalWrite(in3, HIGH);
                      digitalWrite(in4, LOW);
                      delay(del);
              passo = 4;
        break;
      case 2:
            digitalWrite(in1, HIGH);
                      digitalWrite(in2, LOW);
                      digitalWrite(in3, LOW);
                      digitalWrite(in4, HIGH);
                    delay(del);
        passo = 1;
              break;
            case 3:
                      digitalWrite(in1, LOW);
                      digitalWrite(in2, HIGH);
                      digitalWrite(in3, LOW);
                      digitalWrite(in4, HIGH);
                    delay(del);
        passo = 2;
        break;
            case 4:
```

```
            digitalWrite(in1, LOW);
                        digitalWrite(in2, HIGH);
                        digitalWrite(in3, HIGH);
                        digitalWrite(in4, LOW);
                      delay(del);
        passo = 3;
        break;
      }
    }

  }
         return passo;
}


int main(int argc, char **argv)
{

  if(wiringPiSetup()==-1)
  { puts("Deu ruim no setup");
    return -1;
  }

        pinMode(in1, OUTPUT);
        pinMode(in2, OUTPUT);
        pinMode(in3, OUTPUT);
        pinMode(in4, OUTPUT);

  int counter = atoi(argv[1]);
   while(counter!=0)
        {
     passo  = movimenta(calPasso,1);
       counter--;
        }
}
```

APÊNDICE - Implementação do código para acionamento do servo motor

```
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>

int in5 = 5;

void servo0graus()
{
  digitalWrite(in5, HIGH);
  delayMicroseconds(600);
  digitalWrite(in5, LOW);
  for (int i=0; i<32;i++) delayMicroseconds(600);
}

void servoPIgraus()
{
  digitalWrite(in5, HIGH);
  delayMicroseconds(600);
  digitalWrite(in5, LOW);
  for (int i=0; i<26;i++) delayMicroseconds(600);
}


void servo90graus()
{
  digitalWrite(in5, HIGH);
  delayMicroseconds(1500);
  digitalWrite(in5, LOW);
  for (int i=0; i<12;i++) delayMicroseconds(1500);
}
void servo180graus()
{
  digitalWrite(in5, HIGH);
  delayMicroseconds(2400);
```

```
35    digitalWrite(in5, LOW);
36    for (int i=0; i<7;i++) delayMicroseconds(2400);
37  }
38
39  int main()
40  {
41    if(wiringPiSetup()==-1)
42    {
43      puts("Deu ruim no setup");
44      return -1;
45    }
46
47    pinMode(in5, OUTPUT);
48
49      for( int i=0; i<100;i++) servo180graus();
50      delay(500);
51      for(int  i=0; i<100;i++) servoPIgraus();
52    return 0;
53  }
```