



FACULDADE ÚNICA
CIÊNCIA DA COMPUTAÇÃO

INTRODUÇÃO À ORGANIZAÇÃO DE ARQUIVOS E DADOS

TRABALHO PRÁTICO 02
(ESTRUTURAS DE REPETIÇÃO, ESTRUTURAS CONDICIONAIS, VETORES,
SWITCH CASE)

IPATINGA, NOVEMBRO DE 2016

BENAIA RODRIGUES
JOSÉ HENRIQUE INACIO GOMES
LETÍCIA EMANUELA MORAIS SOARES

Trabalho desenvolvido durante a disciplina de
Introdução à Organização de Arquivos e Dados, como
parte da avaliação referente a 2ª etapa.
Professor: Thiago de Medeiros Gualberto

IPATINGA, NOVEMBRO DE 2016

INTRODUÇÃO

Apresentar uma visão geral breve sobre os algoritmos a que se refere este relatório. Descrever os objetivos de desenvolvimento de cada um deles e apresentar as dificuldades apresentadas.

DESENVOLVIMENTO

- **HISTORICO DA LINGUAGEM C**

A linguagem C foi criada em 1972 e implementada em um computador DEC PDP-11 por Dennis Ritchie no Bell Laboratories. C é derivada de duas outras linguagens: Algol 68 e BCPL.

O foco da linguagem C inicialmente foi o desenvolvimento de sistemas operacionais e compiladores. C foi usada com grande êxito na construção de uma nova versão do sistema operacional Unix, que inicialmente foi escrito em Assembly. O grande sucesso obtido no mundo do Unix fez com que a linguagem ganhasse mais e mais adeptos e atualmente, quase todos os grandes sistemas operacionais são construídos em C/C++.

No início dos anos 80 C passou a ser reconhecida como uma linguagem de propósito geral e contava com diversos compiladores desenvolvidos por vários fabricantes. Nessa época havia uma série de compiladores C semelhantes, porém estes compiladores frequentemente apresentavam várias discrepâncias e eram incompatíveis entre si. Isto tornava a padronização da linguagem uma necessidade real. A padronização foi iniciada pela ANSI – American National Standard Institute – em 1983 e finalizada em 1989. Em 1999 foi aprovada uma revisão do padrão anterior e novas funcionalidades foram acrescentadas. Esta revisão ficou conhecida como C99.

A linguagem C é uma linguagem de propósito geral, o que quer dizer que se adapta a praticamente qualquer tipo de projeto, altamente portátil e extremamente rápida em tempo de execução. A linguagem C++ é uma evolução da linguagem C que incorpora orientação a objetos. Linguagens como Java e C# foram influenciadas pela linguagem C.

A linguagem C buscou manter o "contato com o computador real" e ainda sim dar ao programador novas condições para o desenvolvimento de programas em áreas diversas, como comercial, científica e de engenharia.

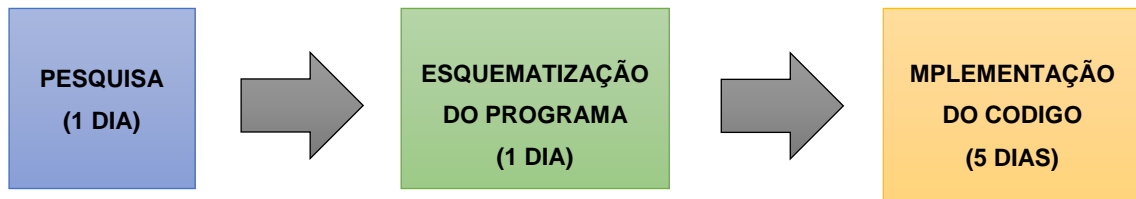
Por muitos anos a sintaxe tida como padrão da linguagem C foi aquela fornecida com o UNIX versão 5.0 do Bell Labs. A principal documentação deste padrão encontra-se na publicação "The C Programming Language", de Brian Kernighan e Dennis Ritchie (K&R), tida como a "bíblia da linguagem C".

O mais interessante desta versão de C era que os programas-fonte criados para rodar em um tipo de computador podiam ser transportados e recompilados em outros sem grandes problemas. A esta característica dá-se o nome de portabilidade. Com ela, uma empresa que desenvolve um programa pode fazê-lo rodar em diferentes computadores sem ter um elevado custo a cada vez que isto for feito.

Em 1985, ANSI (American National Standards Institute) estabeleceu um padrão oficial de C o chamado "C ANSI".

- **FASES DE ELABORAÇÃO**

1. Pesquisa
2. Esquematização do código
3. Implementação do código

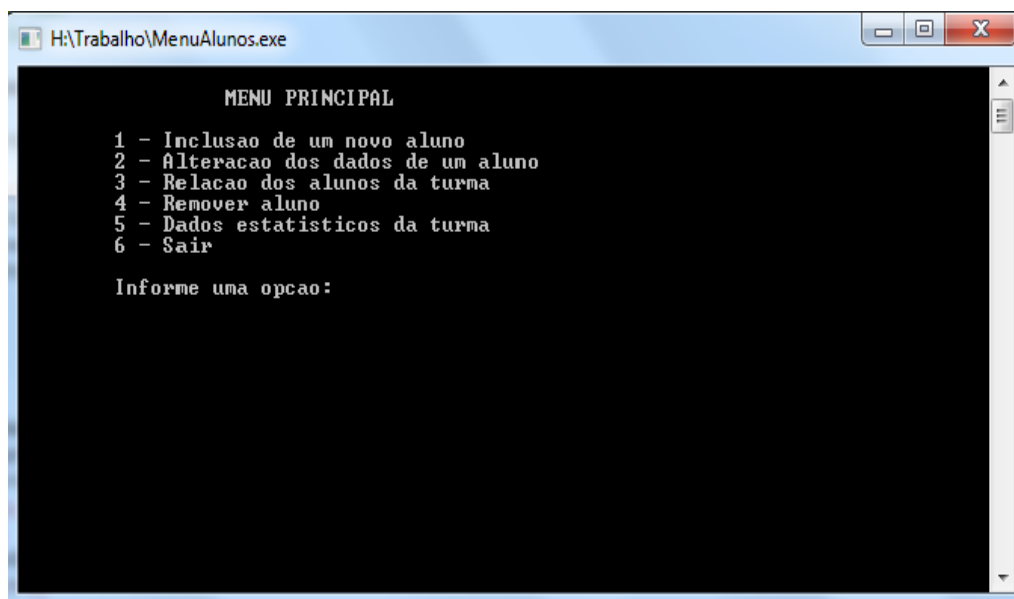


- **EXPLICAÇÃO DO CÓDIGO**

Todo o código foi desenvolvido pelos autores deste relatório. O objetivo do algoritmo (MenuAlunos), baseia-se em simular um menu de uma instituição de ensino, onde serão cadastrados apenas 30 alunos. A lógica utilizada foi bem simples e objetiva. Primeiramente será exibida uma tela para o usuário apresentado todas as opções disponíveis para o funcionamento do menu.

Segue tela e código fonte do menu principal:

```
29
30
31 while(menuprincipal!=6)
32 {
33     system("cls");
34     cout<<"\n\t\t MENU PRINCIPAL  \n\n";
35     cout<<"\t1 - Inclusao de um novo aluno  \n";
36     cout<<"\t2 - Alteracao dos dados de um aluno  \n";
37     cout<<"\t3 - Relacao dos alunos da turma  \n";
38     cout<<"\t4 - Remover aluno  \n";
39     cout<<"\t5 - Dados estatisticos da turma  \n";
40     cout<<"\t6 - Sair  \n";
41
42     cout<<"\n\tInforme uma opcao: ";
43     cin>>menuprincipal;
44
45     system("cls");
```



- **EXPLICAÇÃO DO CÓDIGO**

Opções do Usuário

1. Inclusão de um novo aluno

Essa parte é responsável por fazer o cadastramento do aluno, para isso foi desenvolvido uma estrutura (struct dadospessoais) contendo informações como: matricula, nome, notas das 3 avaliações e quantidade de faltas. Foram usadas também estruturas condicionais (IF-ELSE) para não permitir que os alunos tenham o mesmo número de matricula e também para alertar ao usuário quando for informado uma nota menor que 0 e maior que 10, e quando o número de faltas for maior que 72.

Segue código fonte da struct:

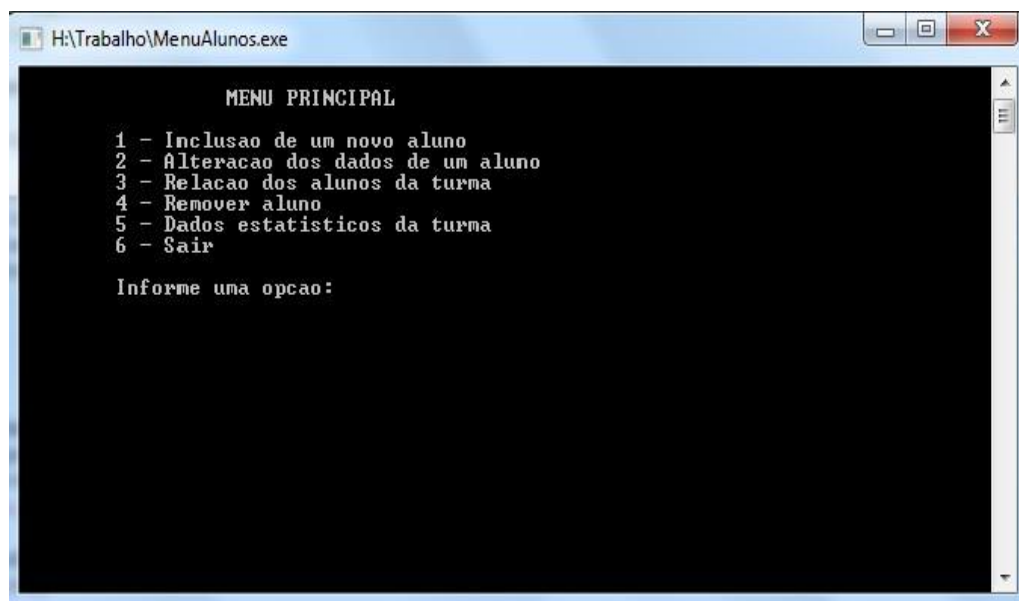
```
8  using namespace std;
9
10 struct dadospessoais
11 {
12     char nome[40];
13     int matricula;
14     int faltas;
15     float nota[3];
16 }
17 };
18
```



```

29
30 while(menuprincipal!=6)
31 {
32     system("cls");
33     cout<<"\n\t\t MENU PRINCIPAL  \n\n";
34     cout<<"\t1 - Inclusao de um novo aluno  \n";
35     cout<<"\t2 - Alteracao dos dados de um aluno  \n";
36     cout<<"\t3 - Relacao dos alunos da turma  \n";
37     cout<<"\t4 - Remover aluno  \n";
38     cout<<"\t5 - Dados estatisticos da turma  \n";
39     cout<<"\t6 - Sair  \n";
40
41     cout<<"\n\tInforme uma opcao: ";
42     cin>>menuprincipal;
43
44     system("cls");
45

```



```

H:\Trabalho\MenuAlunos.exe

MENU PRINCIPAL

1 - Inclusao de um novo aluno
2 - Alteracao dos dados de um aluno
3 - Relacao dos alunos da turma
4 - Remover aluno
5 - Dados estatisticos da turma
6 - Sair

Informe uma opcao:

```

Segue código fonte declarado (linhas 45 á 146):

```
45
46      switch(menuprincipal)
47      {
48          //----- Opcao 1 -----
49
50          case 1:
51
52              x++;
53              cout<<"\tInclusao de um novo aluno  \n";
54              fflush(stdin);
55              cout<<"\n Informe o nome do aluno: ";
56              gets(dados[x].nome);
57              cout<<"\n Informe a matricula do aluno: ";
58              do
59              {
60                  cin>>dados[x].matricula;
61
62
63                  if(dados[x].matricula<=0)
64                  {
65                      cout<<"\n Erro matricula ivalida!!\n Informe um numero maior que 0: ";
66                      cin>>dados[x].matricula;
67                  }
68
69                  v=dados[x].matricula;
70                  for(int cont=0;cont<=x;cont++)
71                  {
72                      if(v==dados[x-1].matricula)
73                      {
74                          t=1;
75                      }
76                      else
77                      {
78                          t=0;
79                      }
80                  }
81
82                  if(t==1)
83                  {
84                      cout<<"\n Matricula informada ja cadastrada, favor informe outra: ";
85                  }
86
87              }while(t!=0);
88
89              cout<<"\n Informe a 1ª nota: ";
90              do
91              {
92                  cin>>dados[x].nota[0];
93
94                  if(dados[x].nota[0]<0 || dados[x].nota[0]>10)
95                  {
96                      cout<<"\nValor invalido\n";
97                      cout<<"Informe um valor entre 0 e 10: ";
98                  }
99
100              }while(dados[x].nota[0]<0 || dados[x].nota[0]>10);
101
```

102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127

```
cout<<"\n Informe a 2ª nota: ";
do
{
    cin>>dados[x].nota[1];

    if(dados[x].nota[1]<0 || dados[x].nota[1]>10)
    {
        cout<<"\nValor invalido\n";
        cout<<"Informe um valor entre 0 e 10: ";
    }

}while(dados[x].nota[1]<0 || dados[x].nota[1]>10);

cout<<"\n Informe a 3ª nota: ";
do
{
    cin>>dados[x].nota[2];

    if(dados[x].nota[2]<0 || dados[x].nota[2]>10)
    {
        cout<<"\nValor invalido\n";
        cout<<"Informe um valor entre 0 e 10: ";
    }

}while(dados[x].nota[2]<0 || dados[x].nota[2]>10);
```

129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146

```
cout<<"\n Informe a quantidade de faltas: ";
do
{
    cin>>dados[x].faltas;
    if(dados[x].faltas<0 || dados[x].faltas>72)
    {
        cout<<"\nValor invalido\n";
        cout<<"Informe um valor entre 0 e 72: ";
    }
}while(dados[x].faltas<0 || dados[x].faltas>72);

cout<<"\n\n Aluno cadastrado com sucesso\n\n";

qtd_alunos = x;

system("pause");
break;
```

2. Alteração dos dados de um aluno

Nesta parte do menu o professor terá acesso aos dados de um certo aluno e poderá fazer alterações em todos os dados do mesmo, exceto no número da matrícula. Neste caso foi criado um sub menu para que o usuário escolha qual opção deseja alterar.

Segue código fonte declarado (linhas 147 á 250):

```
147 //----- Opcao 2 -----
148
149
150 case 2:
151     cout<<"\tAlteracao dos dados de um aluno  \n";
152     cout<<"\n Informe o numero da matricula para alteracao: ";
153     cin>>mat;
154
155     x = qtd_alunos;
156
157     for(int cont=0;cont<=x;cont++)
158     {
159         if(mat==dados[x].matricula)
160         {
161             cout<<"\n Nome do aluno: "<< dados[x].nome;
162             cout<<"\n Matricula do aluno: "<< dados[x].matricula;
163             cout<<"\n 1§ Nota do aluno: "<< dados[x].nota[0];
164             cout<<"\n 2§ Nota do aluno: "<< dados[x].nota[1];
165             cout<<"\n 3§ Nota do aluno: "<< dados[x].nota[2];
166             cout<<"\n Faltas do aluno: "<< dados[x].faltas;
167
168             cout<<"\n\n\t\t ALTERACAO  \n\n";
169             cout<<"\t1 - Alterar o Nome  \n";
170             cout<<"\t2 - Alterar a 1§ Nota  \n";
171             cout<<"\t3 - Alterar a 2§ Nota  \n";
172             cout<<"\t4 - Alterar a 3§ Nota  \n";
173             cout<<"\t5 - Alterar o numero de faltas  \n";
174             cout<<"\t6 - Voltar  \n";
175
176             cout<<"\n\tInforme uma opcao: ";
177             cin>>altera;
178
179             fflush(stdin);
```

181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215

```
switch(altera)
{
    case 1:
        cout<<"\n\t Informe o novo nome: ";
        gets(dados[x].nome);
        break;
    case 2:
        cout<<"\n Informe a 1ª nota: ";
        do
        {
            cin>>dados[x].nota[0];

            if(dados[x].nota[0]<0 || dados[x].nota[0]>10)
            {
                cout<<"\nValor invalido\n";
                cout<<"Informe um valor entre 0 e 10: ";
            }

        }while(dados[x].nota[0]<0 || dados[x].nota[0]>10);
        break;

    case 3:
        cout<<"\n Informe a 2ª nota: ";
        do
        {
            cin>>dados[x].nota[1];

            if(dados[x].nota[1]<0 || dados[x].nota[1]>10)
            {
                cout<<"\nValor invalido\n";
                cout<<"Informe um valor entre 0 e 10: ";
            }

        }while(dados[x].nota[1]<0 || dados[x].nota[1]>10);
        break;
}
```

```

217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250

```

```

case 4:
    cout<<"\n Informe a 3ª nota: ";
    do
    {
        cin>>dados[x].nota[2];

        if(dados[x].nota[2]<0 || dados[x].nota[2]>10)
        {
            cout<<"\nValor invalido\n";
            cout<<"Informe um valor entre 0 e 10: ";
        }

    }while(dados[x].nota[2]<0 || dados[x].nota[2]>10);
    break;

case 5:
    cout<<"\n Informe a quantidade de faltas: ";
    do
    {
        cin>>dados[x].faltas;
        if(dados[x].faltas<0 || dados[x].faltas>72)
        {
            cout<<"\nValor invalido\n";
            cout<<"Informe um valor entre 0 e 72: ";
        }
    }while(dados[x].faltas<0 || dados[x].faltas>72);
    break;
}
break;
}
}

system("pause");
break;

```

```

H:\Trabalho\MenuAlunos.exe

Alteracao dos dados de um aluno

Informe o numero da matricula para alteracao: 1

Nome do aluno: leticia emanuela
Matricula do aluno: 1
1ª Nota do aluno: 5
2ª Nota do aluno: 2
3ª Nota do aluno: 3
Faltas do aluno: 0

ALTERACAO

1 - Alterar o Nome
2 - Alterar a 1ª Nota
3 - Alterar a 2ª Nota
4 - Alterar a 3ª Nota
5 - Alterar o numero de faltas
6 - Voltar

Informe uma opcao:

```

3. Relação dos alunos da turma

Essa parte responsável apenas pela exibição dos dados de todos os alunos em forma de tabela ou individualmente.

Segue o código declarado (linhas 195 á 331):

```
251 //----- Opcao 3 -----
252
253
254 case 3:
255     cout<<"\tRelacao dos alunos da turma  \n";
256     cout<<"\n1 - Listar todos ";
257     cout<<"\n2 - Listar por Aluno ";
258     cout<<"\nEscolha uma das opcoes: ";
259     cin>>op;
260
261     switch(op)
262     {
263     case 1:
264         for(int cont=0;cont<=x;cont++)
265         {
266             media=0;
267
268             media = (dados[cont].nota[0]+dados[cont].nota[1]+dados[cont].nota[2]) / 3;
269
270             cout<<"\n-----";
271             cout<<"\nMatricula:      "<<dados[cont].matricula;
272             cout<<"\nNome:          "<<dados[cont].nome;
273             cout<<"\nFaltas:         "<<dados[cont].faltas;
274             cout<<"\nMedia:          "<<media;
275             if((media>=6)&&(dados[cont].faltas<=18))
276             {
277                 cout<<"\nSituacao:      Aprovado\n";
278             }
279             else
280             if(media<6)
281             {
282                 cout<<"\nSituacao:      Reprovado por nota\n";
283             }
284             else
285             if(dados[cont].faltas>18)
286             {
287                 cout<<"\nSituacao:      Reprovado por frequencia\n";
288             }
289             cout<<"\n-----\n";
290
291             system("pause");
292             break;
293
294 case 2:
295
296     cout<<"Informe a matricula do aluno: ";
297     cin>> mat;
298     media = 0;
299
300     for(int cont=0;cont<=x;cont++)
301     {
302         if(dados[cont].matricula==mat)
303         {
304             media = (dados[cont].nota[0]+dados[cont].nota[1]+dados[cont].nota[2]) / 3;
305             t=1;
306             cout<<"\nMatricula:      "<<dados[cont].matricula;
307             cout<<"\nNome:          "<<dados[cont].nome;
```

```

308 cout<<"\nFaltas:          "<<dados[cont].faltas;
309 cout<<"\nMedia:          "<<media;
310
311 if((media>=6)&&(dados[cont].faltas<=18))
312 {
313     cout<<"\nSituacao:      Aprovado\n";
314 }
315 else
316 if(media<6)
317 {
318     cout<<"\nSituacao:      Reprovado por nota\n";
319 }
320 else
321 if(dados[cont].faltas>18)
322 {
323     cout<<"\nSituacao:      Reprovado por frequencia\n";
324 }
325 }
326 }
327
328 system("pause");
329 break;
330 }
331 break;

```

```

H:\Trabalho\Menu-Alunos-Final (1).exe
Relacao dos alunos da turma
1 - Listar todos
2 - Listar por Aluno
Escolha uma das opcoes: 1

-----
Matricula:      5
Nome:           Leticia Emanuela
Faltas:         0
Media:          10
Situacao:       Aprovado
-----
Matricula:      6
Nome:           Benaia Rodrigues
Faltas:         0
Media:          10
Situacao:       Aprovado
-----
Pressione qualquer tecla para continuar. . .

```


4. Remover aluno

Neste caso, deve ser informado o número de matrícula do aluno. Se existir, deverá ser removido da lista de alunos. Caso contrário, mostrará uma mensagem de “aluno inexistente”.

Segue o código declarado (linhas 331 á 380):

```
331 //----- Opcao 4 -----
332
333 case 4:
334     cout<<"\tRemover aluno  \n";
335
336     cout<<"\n Informe a matricula para remocao: ";
337     cin>>mat;
338
339     for(int cont=0;cont<=x;cont++)
340     {
341         if(mat==dados[cont].matricula)
342         {
343             int aux=x;
344             cout<<"\n Deseja realmente excluir os seguintes dados?";
345             cout<<"\n 1 - Sim";
346             cout<<"\n 2 - Nao";
347
348             cout<<"\n Nome do aluno: "<< dados[cont].nome;
349             cout<<"\n Matricula do aluno: "<< dados[cont].matricula;
350             cout<<"\n 1ª Nota do aluno: "<< dados[cont].nota[0];
351             cout<<"\n 2ª Nota do aluno: "<< dados[cont].nota[1];
352             cout<<"\n 3ª Nota do aluno: "<< dados[cont].nota[2];
353             cout<<"\n Faltas do aluno: "<< dados[cont].faltas;
354             cin>>op;
355
356             switch(op)
357             {
358                 case 1:
359                     for (int cont=0; cont<=x;cont++)
360                     {
361                         strcpy(dados[aux-1].nome,dados[aux].nome);
362                         dados[aux-1].matricula=dados[aux].matricula;
363                         dados[aux-1].nota[0]=dados[aux].nota[0];
364                         dados[aux-1].nota[1]=dados[aux].nota[1];
365                         dados[aux-1].nota[2]=dados[aux].nota[2];
366                         dados[aux-1].faltas=dados[aux].faltas;
367                     }
368                 }
369             }
370             else
371             {
372                 cout<<"Matricula nao encontrada\n\n";
373             }
374             x--;
375             cout<<"Aluno excluido com sucesso.\n\n";
376             system("pause");
377             break;
378
379
380
381
```

5. Dados estatísticos da turma

Deve-se apresentar a média geral das notas finais dos alunos da turma, média geral dos números de faltas dos alunos da turma, matrícula e nome do aluno de maior nota final da turma, juntamente com tal nota. Deve ser apresentado se tiver mais de 5 alunos cadastrado.

6. Sair

Neste caso ao informar o número 6 o programa ira finalizar sua execução.

Segue código fonte declarado (linhas):

```
260  
261  
262     case 6:  
263         return 0;  
264         break;  
265  
266     default:  
267         cout<<"\n OPCAO INVALIDA..  \n\n\n";  
268         system("pause");  
269         break;  
270 }  
---
```

- **DIFICULDADES ULTRAPASSADAS**

As dificuldades foram muitas, mas com o devido esforço foram ficando para trás. Foram notadas dificuldades para mover o conteúdo de uma posição para outra em um vetor e também para remover a última posição.

Para superar as dificuldades foram pesquisados vários métodos para remoção de dados em um vetor e além de pesquisas algumas dúvidas foram tiradas com profissionais da área e até mesmo com alguns alunos do curso.

CONCLUSÃO

O desenvolvimento deste trabalho gerou novos conhecimentos. A linguagem C é muito extensa e complexa, o trabalho requereu uma constante pesquisa de material para satisfazer as dificuldades que iam aparecendo. De acordo com seu desenvolvimento, o trabalho foi ganhando forma e, como consequência, foi necessário cada vez mais trabalho de pesquisa, e com isso fomos aprendendo cada vez mais sobre a linguagem.