

UNIVERSIDADE PAULISTA - UNIP
PROJETO INTEGRADO MULTIDISCIPLINAR
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
ARTHUR PEREIRA (2037060)

PROJETO MAIS SAÚDE

São Paulo - Morumbi
Novembro de 2020

ARTHUR PEREIRA - 2037060

PROJETO MAIS SAÚDE

Projeto Integrado Multidisciplinar para a obtenção do título de graduação em Análise e Desenvolvimento de Sistemas, apresentado à Universidade Paulista – UNIP EaD.

São Paulo - Morumbi

Novembro de 2020

RESUMO

Tendo em vista a pandemia de Sars-COV-2 (Covid-19), a tecnologia se fez uma grande aliada para o combate e prevenção do surto da doença. Neste Projeto Integrado Multidisciplinar, o PIM, foi feito um exemplo de uma ótima interface que pode auxiliar pessoas da área da saúde a continuar, bravamente, a combater o surto.

Palavras-chave: Saúde, prevenção, combate, Linguagem C.

ABSTRACT

In view of the Sars-VOC-2 (Covid-19) pandemic, technology has become a great ally for combating and preventing the outbreak of the disease. In this Integrated Multidisciplinary Project, PIM, an example of a great interface was made that can help people in the health area to continue, bravely, to fight the outbreak.

Keywords: Health, prevention, combat, C Code.

SUMÁRIO

RESUMO	4
ABSTRACT	5
SUMÁRIO	6
INTRODUÇÃO	7
DESENVOLVIMENTO	8
CONCLUSÃO	17
REFERÊNCIAS	18

INTRODUÇÃO

Independente do que o ser humano pauta em sua vida como “marco-zero”, do Big-Bang ao Criacionismo. O marco de maior relevância na vida humana é o ato de existir a comunhão entre os seres vivos. De Adão e Eva aos Australopithecus, a koinonia, palavra grega para a comunhão, é o que diferencia as sociedades, o grau de comunhão, ou interligação, entre os seus agentes, demonstram o grau de desenvolvimento da civilização. Os indígenas centro-americanos se destacaram mais que os sul americanos pelo fato da unidade.

Agora, no século XXI, em que a sociedade já não é mais segregada em tribos, mas é global. A comunhão e solidariedade se destacam sobre as relações interpessoais, e com isso o avanço tecnológico pode acender-se ao que Klaus Schwab chama de quarta revolução industrial. O mundo globalizado revela que profissões marginalizadas como entregadores de fast-food podem assumir o protagonismo em tempos de crise, como a vida em 2020 por conta da pandemia de Covid-19.

Sendo assim, a sociedade alavancou-se em níveis maiores para que haja de modo íntegro uma intersecção nas diversas áreas do saber. Por este motivo, a área da tecnologia uniu-se com a área da saúde para trazer soluções para combater e prevenir o surto de Covid.

DESENVOLVIMENTO

1. PROPOSTA TÉCNICA

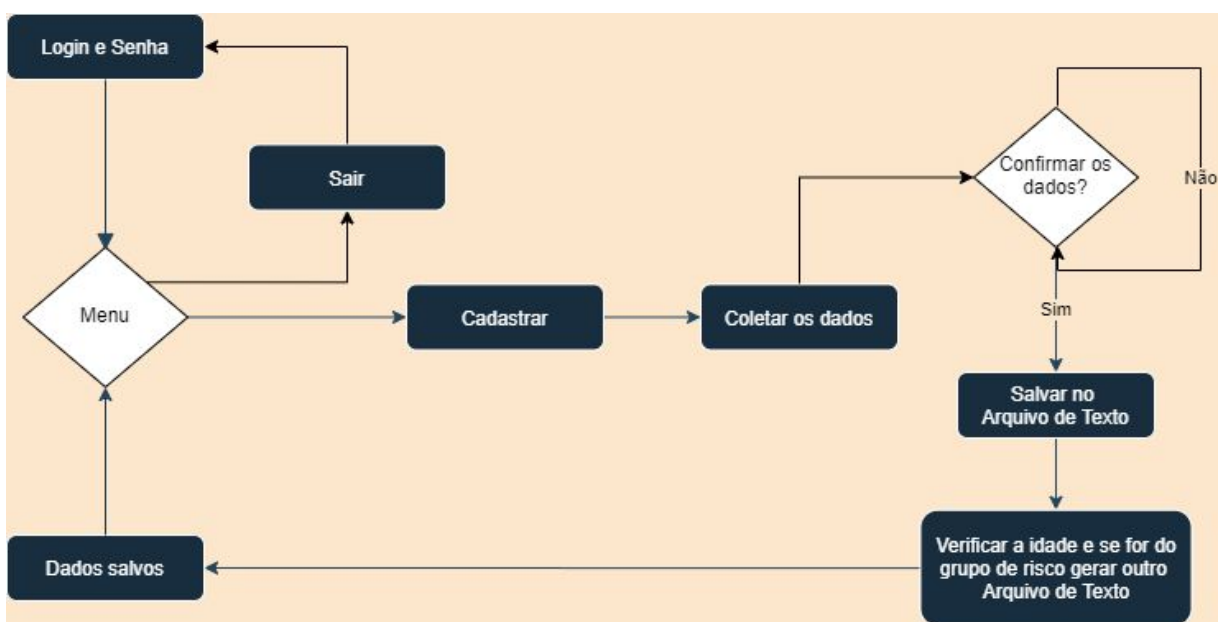
1.1 Objetivo

A proposta neste projeto integrado multidisciplinar, o PIM, mais do que uma mera codificação em linguagem C, é uma forma de estudantes da área da tecnologia alavancarem-se em um serviço para a comunidade global e auxiliar no combate da Covid-19.

O Projeto foi denominado pelo codificador, Arthur Pereira, como “Mais Saúde”. Sendo um projeto em que tem como função cadastrar pessoas que foram contaminadas pelo vírus e elencar em um arquivo de texto, que pode ser aplicado a hospitais, UBS ou AMAs.

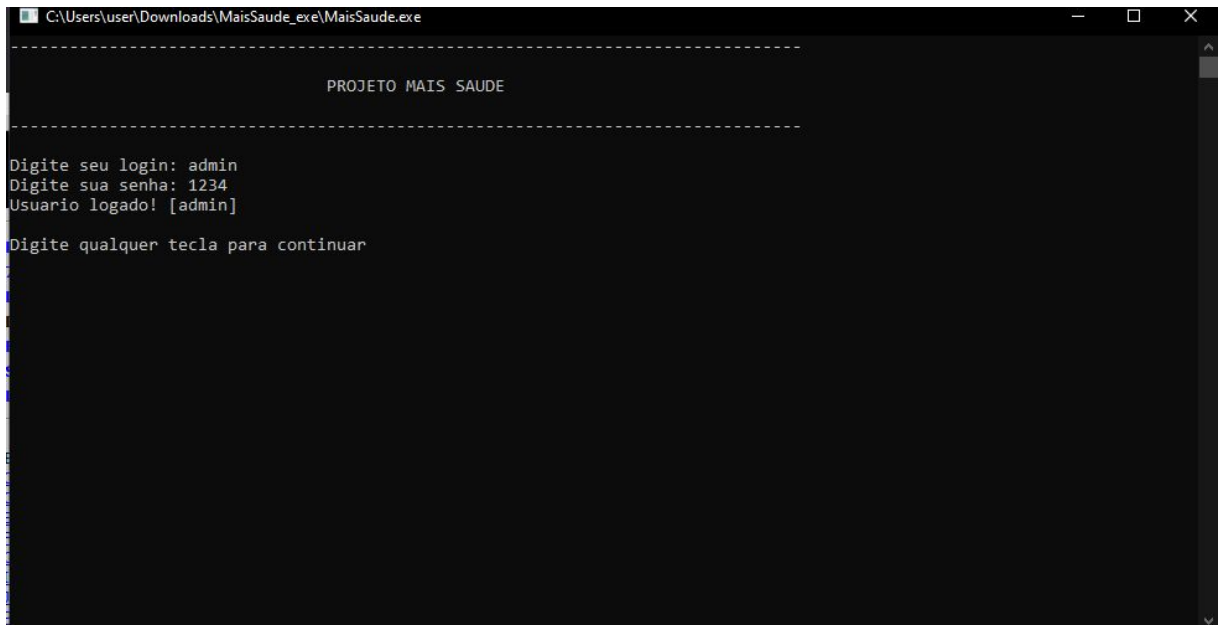
1.2 Lógica

O caminho lógico está descrito no Fluxograma, desenhado em parceria com a designer Giovanna Berson.



1.3 Telas do programa:

► Tela de Login:



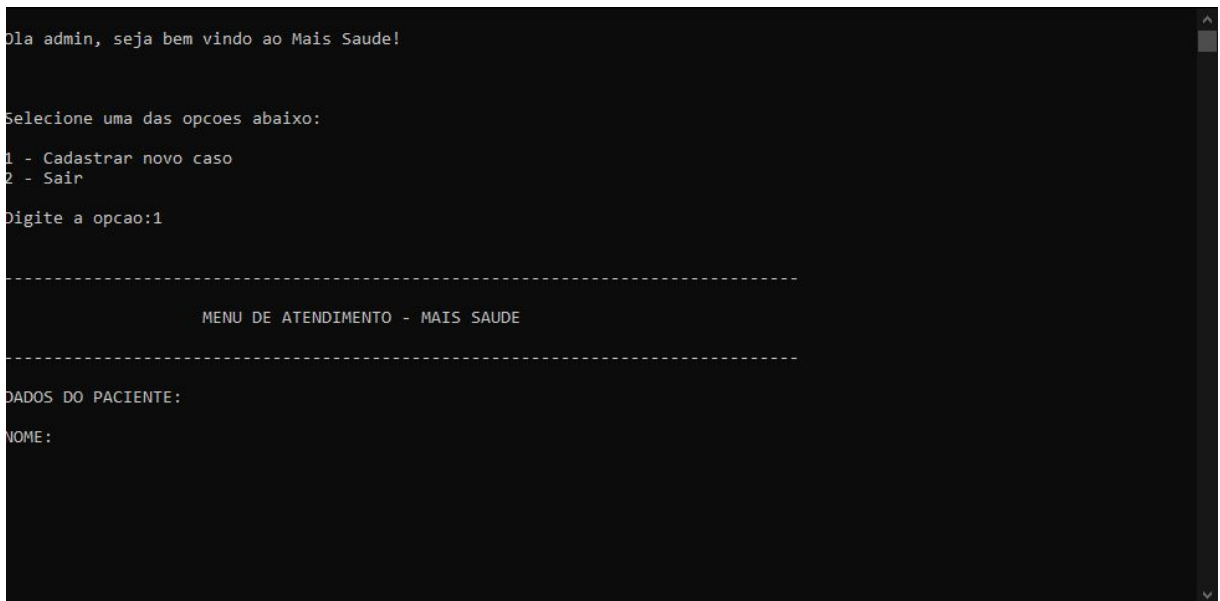
```
C:\Users\user\Downloads\MaisSaude_exe\MaisSaude.exe

-----
                        PROJETO MAIS SAUDE
-----

Digite seu login: admin
Digite sua senha: 1234
Usuario logado! [admin]

Digite qualquer tecla para continuar
```

► Tela de menu:



```
Ola admin, seja bem vindo ao Mais Saude!

Selecione uma das opcoes abaixo:
1 - Cadastrar novo caso
2 - Sair

Digite a opcao:1

-----
                        MENU DE ATENDIMENTO - MAIS SAUDE
-----

DADOS DO PACIENTE:
NOME:
```

► Tela de coleta de dados:

```
-----
MENU DE ATENDIMENTO - MAIS SAUDE
-----

DADOS DO PACIENTE:

NOME: Arthur Pereira
CPF: 450.183.538-99
TELEFONE: (11)97719-1778
ENDERECO: Rua Ernest Renam
NUMERO: 605
COMPLEMENTO: Bloco 2 Apartamento 44
BAIRRO: Fazzenda Morumbi
CIDADE: São PAulo
ESTADO (SIGLA): SP
CEP: 05656-105
DATA DE NASCIMENTO (DD/MM/AAAA): 03/02/2001
EMAIL: arthurpe44@gmail.com
DIAGNOSTICO (DATA - DD/MM): 03/02
COMORBIDADES

1 - DIABETES
2 - OBESIDADE
3 - HIPERTENSAO
4 - TUBERCULOSE
9 - OUTROS

0 - NAO POSSUI COMORBIDADES
```



► Tela de Confirmar os dados:

```
-----
OS DADOS A CIMA ESTAO CORRETOS? DESEJA CONFIRMAR?
-----

1 - SIM
2 - NAO

Digite a opcao:1_
```

► Tela de salvar dados do grupo de risco:

Nome	Data de modificação	Tipo	Tamanho
 DadosDoGrupoDeRisco	17/11/2020 00:01	Documento de Te...	1 KB
 DadosDoPaciente	17/11/2020 00:01	Documento de Te...	1 KB

► Tela de retorno ao menu:

```
-----
Selecione uma das opcoes abaixo:
-----

1 - Cadastrar novo caso
2 - Sair

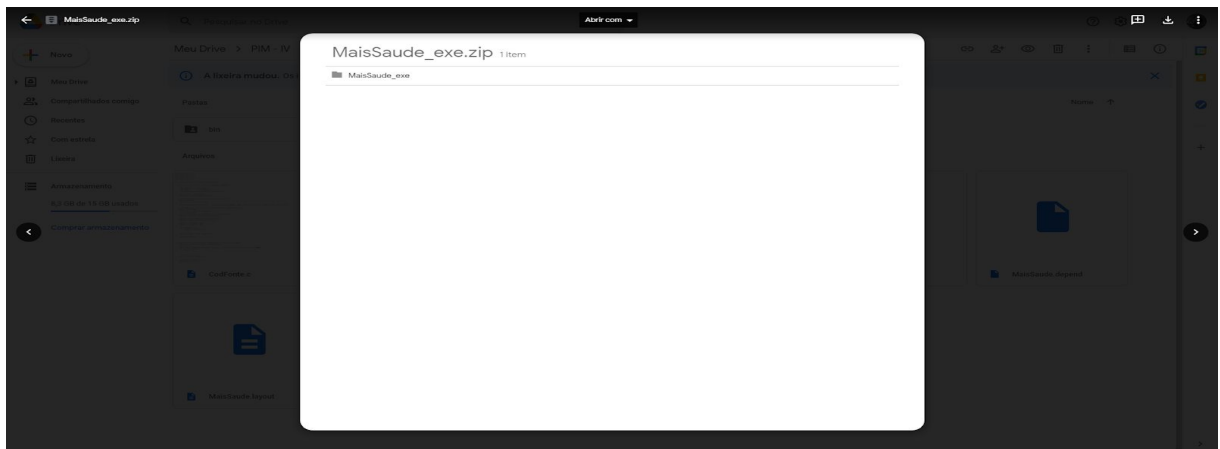
Digite a opcao:_
```

1.4 Manual de instalação

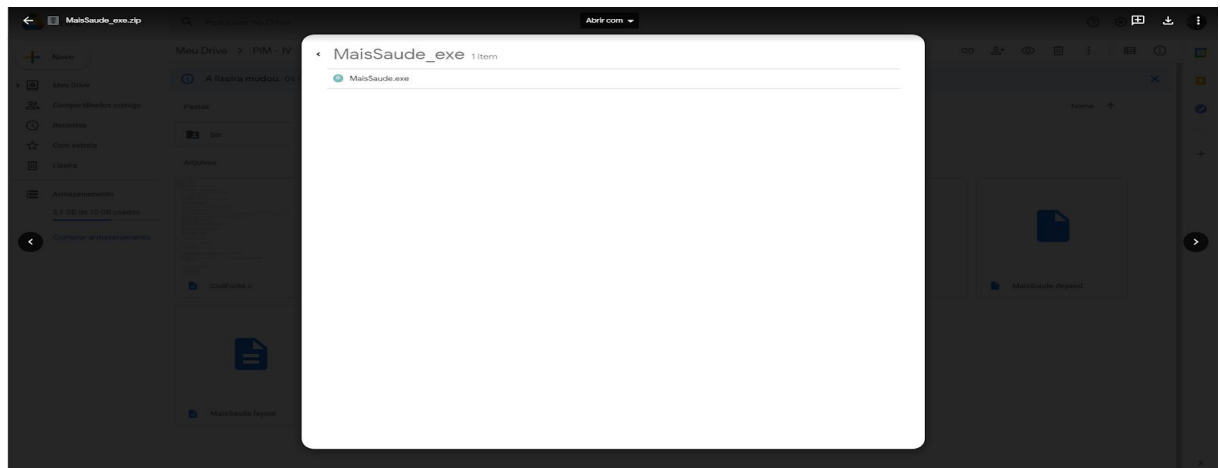
No arquivo: MaisSaude.exe, na pasta comprimida MaisSaude_exe basta clicar duas vezes, fazer a instalação do programa e começar a fazer o uso.

Conforme passo a passo em prints:

- Abra o arquivo em .zip:



- Abra a pasta dentro do arquivo em .zip:



- Depois, descompacte o arquivo com um algum descompactador (WinRar, 7-zip) e baixe o executável (.exe):

← MaisSaude_exe 1 item

➊ MaisSaude.exe

2. Como compilar o programa:

Foi selecionada as bibliotecas:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <conio.h>
4 #include <string.h>
5 #include <time.h>
6
```

Para ser necessário usar o programa.

- Foi usado uma codificação específica para Calcular a idade, transferindo a data de nascimento (em caracteres para números inteiros):

(Prints do CODE::Block)

```
7 int calculaIdade(const char* nascimento){
8
9     int ano, mes, anoAtu, mesAtu, anoDiff, mesDiff;
10
11     char data[] = "01/01/2016";
12     char hora[] = " - 00:00:00";
13     char *datahora=(char*)malloc(1*sizeof(char));
14
15     strcpy(data, nascimento);
16     datahora = strcat(data, hora);
17
18     //contendo para struct que armazena data e hora
19     struct tm tm;
20     struct tm *data_hora_atual;
21
22     sscanf(datahora, "%d/%d/%d - %d:%d:%d", &tm.tm_mday, &mes, &ano, &tm.tm_hour, &tm.tm_min, &tm.tm_sec);
23
24     //transforma do time_t para struct o tempo em segundos
25     time_t segundos;
26     //obtem o tempo em segundos
27     time(&segundos);
28     //para converter de segundos para o tempo local
29     data_hora_atual = localtime(&segundos);
30
31     anoAtu = data_hora_atual->tm_year+1900;
32     mesAtu = data_hora_atual->tm_mon+1;
33
34     mesDiff = mesAtu - mes;
35     anoDiff = anoAtu - ano;
36
37     if(mesDiff < 0){
38         anoDiff = anoDiff - 1;
39     }
40
41     printf("Idade: %i\n", anoDiff);
42
43     return anoDiff;
44 }
45
```

- Outra para gravar os arquivos:

```
44 }
45
46 int gravarArquivo(const char* nomeArquivo, const char* linha){
47     char *Str=(char*)malloc(1*sizeof(char));
48     int result;
49     FILE *arq = fopen(nomeArquivo, "a"); // cria um arquivo novo para gravar
50     if (arq == NULL){
51         return 0;
52     }
53
54     /* grava a linha */
55     result = fputs(linha, arq);
56
57     /* grava a linha */
58     strcpy(Str, "\n");
59     result = fputs(Str, arq);
60
61     if (result == EOF)
62         return 0;
63     fclose(arq);
64     return 1;
65 }
66
```

- Outro para evitar quebra de linhas (no arquivo TXT):

```

67 char* replaceWord(const char* s, const char* oldW, const char* newW)
68 {
69     char* result;
70     int i, cnt = 0;
71     int newWlen = strlen(newW);
72     int oldWlen = strlen(oldW);
73
74     // Counting the number of times old word
75     // occurs in the string
76     for (i = 0; s[i] != '\0'; i++) {
77         if (strstr(s+i, oldW) == s+i) {
78             cnt++;
79             // Jumping to index after the old word.
80             i += oldWlen - 1;
81         }
82     }
83
84     // Making new string of enough length
85     result = (char*)malloc(i + cnt * (newWlen - oldWlen) + 1);
86
87     i = 0;
88     while (*s) {
89         // Compare the substring with the result
90         if (strstr(s, oldW) == s) {
91             strcpy(result+i, newW);
92             i += newWlen;
93             s += oldWlen;
94         }
95         else {
96             result[i++] = *s++;
97         }
98     }
99
100     result[i] = '\0';
101     return result;
102 }
103
104

```

- Outros secundários para limpar buffer do teclado e limpar a tela:

```

102     return result;
103 }
104
105 void limparbuffer(void) // Função utilizada para limpar o buffer do teclado
106 {
107     int c;
108     while ( (c = getchar()) != '\n' && c != EOF ) { }
109 }
110
111 void clear(void)
112 {
113     while ( getchar() != '\n' );
114 }
115

```

- Código principal:

```

CodFonte.c - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools Plugins DoxyBlocks Settings Help
global> replaceWord(const char* s, const char* oldW, const char* newW): char
Management
Projects Symbols Files
Workspace
115 int main()
116 {
117     int userTentativas;
118     char userInput[30];
119     char passInput[30];
120
121     char *menuTendimento=(char*)malloc(1*sizeof(char));
122     char *nomePaciente=(char*)malloc(1*sizeof(char));
123     char *cpfPaciente=(char*)malloc(1*sizeof(char));
124     char *telPaciente=(char*)malloc(1*sizeof(char));
125     char *endRuaPaciente=(char*)malloc(1*sizeof(char));
126     char *endBairroPaciente=(char*)malloc(1*sizeof(char));
127     char *endCidadePaciente=(char*)malloc(1*sizeof(char));
128     char *endEstadoPaciente=(char*)malloc(1*sizeof(char));
129     char *endCEP=(char*)malloc(1*sizeof(char));
130     char *dataDiagPaciente=(char*)malloc(1*sizeof(char));
131     char *optAgrevPaciente=(char*)malloc(1*sizeof(char));
132     char *optConvPaciente=(char*)malloc(1*sizeof(char));
133     char *nascPaciente=(char*)malloc(1*sizeof(char));
134     char *emailPaciente=(char*)malloc(1*sizeof(char));
135     char *dataDiagPaciente=(char*)malloc(1*sizeof(char));
136     char *optAgrevPaciente=(char*)malloc(1*sizeof(char));
137     char *optConvPaciente=(char*)malloc(1*sizeof(char));
138     char *confirmCadast=(char*)malloc(1*sizeof(char));
139
140     char *dadosPaciente=(char*)malloc(1*sizeof(char));
141     char *dadosComorbido=(char*)malloc(1*sizeof(char));
142
143     userTentativas = 1;
144     // userTentativas = 0; // para teste
145     char user[6] = "admin";
146     char pass[6] = "1234";
147     printf("-----\n\n");
148     printf("PROJETO MAIS SAUDE\n\n");
149     printf("-----\n\n");
150     while (userTentativas > 0) {
151         // para teste
152         printf("Digite seu login: ");
153         gets(userInput);

```

```

CodFonte.c - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Plugins DDD/Blocks Settings Help
<global>
man0: int
Management
Projects Symbols Files
Workspace
Start here CodFonte.c
152 printf("Digite seu login: ");
153 gets(userInput);
154 /* nome e senha */
155 printf("Digite sua senha: ");
156 gets(passInput);
157
158 /* verifica o usuario e senha */
159 if(strcmp(userInput, user) == 0 && strcmp(passInput, pass) == 0){
160     printf("Usuario logado! [%s] \n\n", userInput);
161     printf("Digite qualquer tecla para continuar \n\n");
162     getch();
163     break;
164 } else{
165     /* se o numero de tentativas de login for == 4, sair do programa */
166     if(userTentativas == 4){
167         printf("Excedeu o limite de tentativas \n");
168         printf("Digite qualquer tecla para continuar \n\n");
169         getch();
170         exit(0);
171     } else{ /* informa o erro e pede para tentar novamente */
172         printf("Usuario e senha invalidos! [%s] \n", userInput);
173         printf("Digite qualquer tecla para continuar \n\n");
174         getch();
175     }
176     /* soma as tentativas */
177     userTentativas = userTentativas + 1;
178 }
179
180 /* boas vindas */
181 printf("Ola %s, seja bem vindo ao Mais Saude! \n\n", userInput);
182
183 menu: printf("\n\nSelecione uma das opcoes abaixo: \n\n");
184 printf("1 - Cadastrar novo caso \n");
185 printf("2 - Sair \n\n");
186
187 printf("Digite a opcao:");
188 gets(menuAtendimento);
189
190

```

C:\Users\User\Downloads\CodFonte.c Windows (CR+LF) WINDOWS-1252 Line 189, Column 25 Insert Read/Write default

```

CodFonte.c - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Plugins DDD/Blocks Settings Help
<global>
man0: int
Management
Projects Symbols Files
Workspace
Start here CodFonte.c
191 // printf("Menu Selecionado: %i, Obrigado \n\n", menuAtendimento);
192 // getch();
193
194 if( strcmp(menuAtendimento, "2") == 0){
195     printf("Obrigado! \n\nDigite qualquer tecla para sair.");
196     getch();
197     exit(0);
198 }else{
199     cadastro: printf("\n\n-----\n\n");
200     printf("          MENU DE ATENDIMENTO - MAIS SAUDE\n\n");
201     printf("-----\n\n");
202     printf("DADOS DO PACIENTE: \n\n");
203
204     printf("NOME: ");
205     gets(nomePaciente);
206
207     printf("CPF: ");
208     gets(cpfPaciente);
209
210     printf("TELEFONE: ");
211     gets(telPaciente);
212
213     printf("ENDERECO: ");
214     gets(endRuaPaciente);
215
216     printf("NUMERO: ");
217     gets(endNumPaciente);
218
219     printf("COMPLEMENTO: ");
220     gets(endCompPaciente);
221
222     printf("BAIRRO: ");
223     gets(endBairroPaciente);
224
225     printf("CIDADE: ");
226     gets(endCidadePaciente);
227
228     printf("ESTADO (SIGLA): ");
229     gets(endEstPaciente);

```

C:\Users\User\Downloads\CodFonte.c Windows (CR+LF) WINDOWS-1252 Line 205, Column 28 Insert Read/Write default

```
CodFonte.c - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DmgBlocks Settings Help
<global>
main() : int
Management
Projects Symbols Files
Workspace
Start here CodFonte.c
231 printf("CEP: ");
232 gets(endCepPaciente);
233
234 printf("DATA DE NASCIMENTO (DD/MM/AAAA): ");
235 gets(nascPaciente);
236
237 printf("EMAIL: ");
238 gets(emailPaciente);
239
240 printf("DIAGNOSTICO (DATA - DD/MM): ");
241 gets(dataDiagPaciente);
242
243 printf("COMORBIDADES \n\n");
244 printf("1 - DIABETES \n");
245 printf("2 - OBESIDADE \n");
246 printf("3 - HIPERTENSAO \n");
247 printf("4 - TUBERCULOSE \n");
248 printf("5 - OUTROS \n\n");
249 printf("0 - NAO POSSUI COMORBIDADES \n\n");
250 /* CRIANDO int codecl_MINGW_NOTHROW printf(const char*, ...) */
251 printf("SELECIONE UMA DAS OPCOES: ");
252 gets(optAgravPaciente);
253
254 optAgravPaciente = replaceWord(optAgravPaciente, "\n", "");
255
256 /* controla primeiro atribuido o valor de agravPaciente diretamente, depois substitui pela função strcmp */
257 if( strcmp(optAgravPaciente, "1") == 0 ){
258     agravPaciente = strcpy(agravPaciente, "DIABETES");
259 } else if( strcmp(optAgravPaciente, "2") == 0 ){
260     agravPaciente = strcpy(agravPaciente, "OBESIDADE");
261 } else if( strcmp(optAgravPaciente, "3") == 0 ){
262     agravPaciente = strcpy(agravPaciente, "HIPERTENSAO");
263 } else if( strcmp(optAgravPaciente, "4") == 0 ){
264     agravPaciente = strcpy(agravPaciente, "TUBERCULOSE");
265 } else if( strcmp(optAgravPaciente, "5") == 0 ){
266     agravPaciente = strcpy(agravPaciente, "OUTROS");
267 } else {
268     agravPaciente = strcpy(agravPaciente, "SEM COMORBIDADES");
269 }
```

C:\Users\User\Downloads\CodFonte.c Windows (CR-LF) WINDOWS-1252 Line 249, Column 31 Insert Read/Write default

```
CodFonte.c - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DmgBlocks Settings Help
<global>
main() : int
Management
Projects Symbols Files
Workspace
Start here CodFonte.c
271 nomePaciente = replaceWord(nomePaciente, "\n", "");
272 cpfPaciente = replaceWord(cpfPaciente, "\n", "");
273 telPaciente = replaceWord(telPaciente, "\n", "");
274 endRuaPaciente = replaceWord(endRuaPaciente, "\n", "");
275 endNumPaciente = replaceWord(endNumPaciente, "\n", "");
276 endCompPaciente = replaceWord(endCompPaciente, "\n", "");
277 endBairroPaciente = replaceWord(endBairroPaciente, "\n", "");
278 endCidadePaciente = replaceWord(endCidadePaciente, "\n", "");
279 endEstPaciente = replaceWord(endEstPaciente, "\n", "");
280 endCepPaciente = replaceWord(endCepPaciente, "\n", "");
281 emailPaciente = replaceWord(emailPaciente, "\n", "");
282 dataDiagPaciente = replaceWord(dataDiagPaciente, "\n", "");
283 agravPaciente = replaceWord(agravPaciente, "\n", "");
284 nascPaciente = replaceWord(nascPaciente, "\n", "");
285
286 printf("-----\n\n");
287 printf("OS DADOS A CIMA ESTAO CORRETOS? DESEJA CONFIRMAR? \n\n");
288 printf("1 - SIM \n");
289 printf("2 - NAO \n\n");
290 printf("DIGITE 1 ou 2:");
291
292 gets(confirmCadastr);
293
294 if( strcmp(confirmCadastr, "2") == 0 ){
295     goto cadastro;
296 } else {
297     printf("-----\n\n");
298 }
299
300 dadosPaciente = strcpy(dadosPaciente, "");
301 dadosPaciente = strcat(dadosPaciente, nomePaciente);
302 dadosPaciente = strcat(dadosPaciente, ";");
303 dadosPaciente = strcat(dadosPaciente, cpfPaciente);
304 dadosPaciente = strcat(dadosPaciente, ";");
305 dadosPaciente = strcat(dadosPaciente, telPaciente);
306 dadosPaciente = strcat(dadosPaciente, ";");
307 dadosPaciente = strcat(dadosPaciente, endRuaPaciente);
308 dadosPaciente = strcat(dadosPaciente, ";");
309 dadosPaciente = strcat(dadosPaciente, "\n");
```

C:\Users\User\Downloads\CodFonte.c Windows (CR-LF) WINDOWS-1252 Line 308, Column 63 Insert Read/Write default

```

CodFonte.c - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools- Plugins DmgBlocks Settings Help
<global>
main() {int

Management
Projects Symbols Files
Workspace

309 dadosPaciente = strcat(dadosPaciente, ",");
310 dadosPaciente = strcat(dadosPaciente, endNumPaciente);}
311 dadosPaciente = strcat(dadosPaciente, ",");
312 dadosPaciente = strcat(dadosPaciente, endCompPaciente);
313 dadosPaciente = strcat(dadosPaciente, ",");
314 dadosPaciente = strcat(dadosPaciente, endBairroPaciente);
315 dadosPaciente = strcat(dadosPaciente, ",");
316 dadosPaciente = strcat(dadosPaciente, endCidadePaciente);
317 dadosPaciente = strcat(dadosPaciente, ",");
318 dadosPaciente = strcat(dadosPaciente, endEstPaciente);
319 dadosPaciente = strcat(dadosPaciente, ",");
320 dadosPaciente = strcat(dadosPaciente, endCepPaciente);
321 dadosPaciente = strcat(dadosPaciente, ",");
322 dadosPaciente = strcat(dadosPaciente, nascPaciente);
323 dadosPaciente = strcat(dadosPaciente, ",");
324 dadosPaciente = strcat(dadosPaciente, emailPaciente);
325 dadosPaciente = strcat(dadosPaciente, ",");
326 dadosPaciente = strcat(dadosPaciente, dataDiagPaciente);
327 dadosPaciente = strcat(dadosPaciente, ",");
328 dadosPaciente = strcat(dadosPaciente, agravPaciente);
329 dadosPaciente = strcat(dadosPaciente, "\n");
330
331 gravarArquivo("DadosDoPaciente.txt", dadosPaciente);
332
333 getch();
334
335 int idadePaciente;
336 char *strIdadePaciente=(char*)malloc(1*sizeof(char));
337 idadePaciente = calculaIdade(nascPaciente);
338
339 if(idadePaciente >= 65 && strcmp(agravPaciente, "SEM_COMORBIDADES") != 0){
340     //AVANÇADA("ELIGIVEL");
341
342     dadosComorbido = strcpy(dadosComorbido, "");
343     dadosComorbido = strcat(dadosComorbido, endCepPaciente);
344     dadosComorbido = strcat(dadosComorbido, ",");
345     dadosComorbido = strcat(dadosComorbido, ",");
346
347     int length = sprintf( NULL, 0, "%d", idadePaciente);

```

```

344     dadosComorbido = strcat(dadosComorbido, endCepPaciente);
345     dadosComorbido = strcat(dadosComorbido, ",");
346
347     int length = sprintf( NULL, 0, "%d", idadePaciente);
348     sprintf(strIdadePaciente, length + 1, "%d", idadePaciente);
349
350     dadosComorbido = strcat(dadosComorbido, strIdadePaciente);
351     dadosComorbido = strcat(dadosComorbido, "\n");
352
353     gravarArquivo("DadosDoSupondoRisco.txt", dadosComorbido);
354 }
355 system("CLS");
356 goto menu;
357
358 printf("%i", nascPaciente);
359 getch();
360
361 }
362
363 return 0;
364
365 }
366

```


3. Como testar?

Para fazer o teste é necessário usar o Login(admin) e a senha(1234). Caso erre, a pessoa terá mais três chances de preencher corretamente, totalizando quatro chances.

Assim que fizer o login o cliente irá escolher a opção “Cadastrar” ou “Sair”, caso queira sair, a aplicação irá pedir uma tecla para encerrar. Caso o cliente clique em cadastrar, irá para a próxima tela.

Na próxima tela irá para a coleta de dados e um menu para selecionar as comorbidades, após selecionar a última opção terá dois caminhos “Confirmar” ou “Não confirmar” os dados. Ao selecionar não confirmar, irá retornar para o menu, ao confirmar irá para outro processo, o de salvar o TXT.

Usando uma codificação para fazer cálculo com data e retornar a idade da pessoa. Ao verificar a idade, se for acima de 65 anos de idade e tiver alguma comorbidade irá salvar em mais outro arquivo de texto, .txt.

Após salvar os dados o cliente retornará para o menu e terá a oportunidade de refazer o processo.

CONCLUSÃO

Conforme desenhado o projeto, em linguagem C, será entregue os binários do programa e o código fonte para que os clientes consigam personalizar e alterar conforme a preferência e necessidades específicas da realidade do cliente.

Neste projeto cabe a parabenização e auxílio de Giovanna Berson que foi a designer e de Melkih W. de Oliveira que foi o supervisor do codificador Arthur Pereira.

Cabe, aqui, concluir que a Universidade Paulista foi a responsável por fomentar em seus estudantes a paixão pelo desenvolvimento e serviço a sociedade.

REFERÊNCIAS

- <http://devfuria.com.br/logica-de-programacao/strings-concatenar>
- <http://www.facom.ufu.br/~backes/gsi011/Aula00-Arquivos.pdf>
- https://www.techonthenet.com/c_language/standard_library_functions/string_h/strcat.php
- <https://www.programiz.com/c-programming/c-switch-case-statement>
- <https://www.youtube.com/playlist?list=PLesCEcYj003SwVdufCQM5FIbrOd0G1M4>
- Ilustrações : Giovanna Berson (Designer Gráfica) - <https://www.linkedin.com/in/giovanna-berson-175b39181/>
- Supervisão: Melkih Washington (Gestor Sênior em Plataformas Digitais e Inovações) - <https://www.linkedin.com/in/melkih/>

Regras e Normas da ABNT 2012 para formatação de trabalhos acadêmicos.

Trabalhos ABNT. Disponível em:

<<http://www.trabalhosabnt.com/regras-normas-abnt-formatacao>>. Acesso em 09 de Outubro de 2020.

Trabalhos acadêmicos: Normas da ABNT. Firb. Disponível em:

<<http://www.firb.br/abntmonograf.htm>>. Acesso em 04 de abr. de 2012.