

Vending Machine - PMR3402

Rafael Sobral Augusto - NUSP 10337193

2020

1 Diagramas UML

1.1 Casos de uso

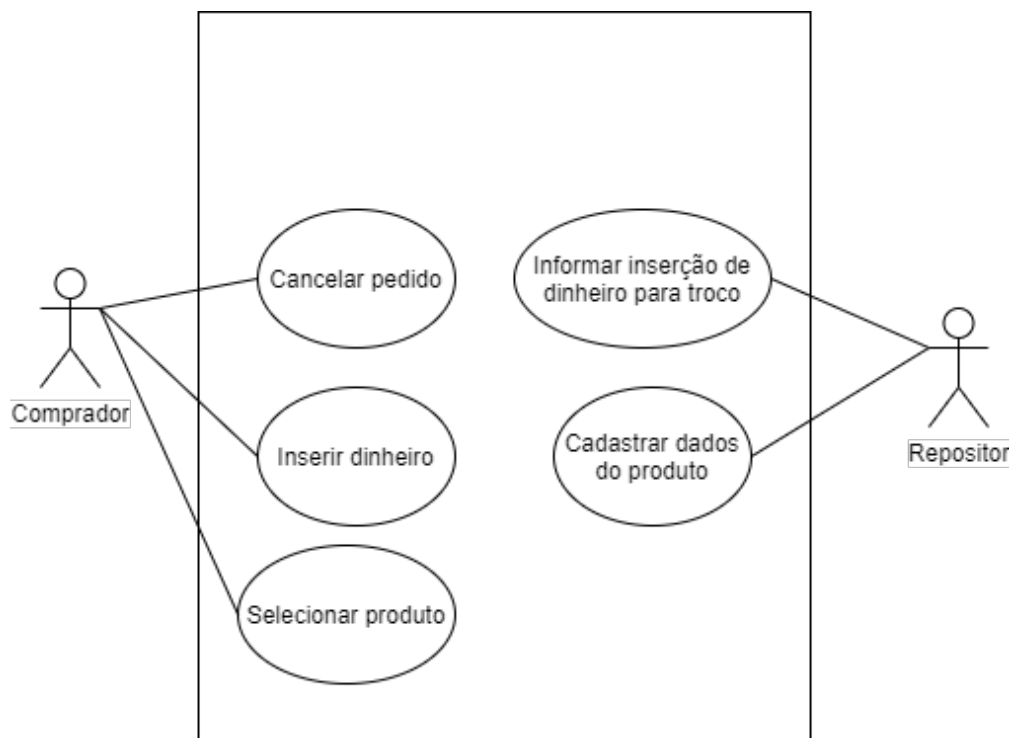


Figura 1: Casos de uso.

1.2 Máquina de Estados

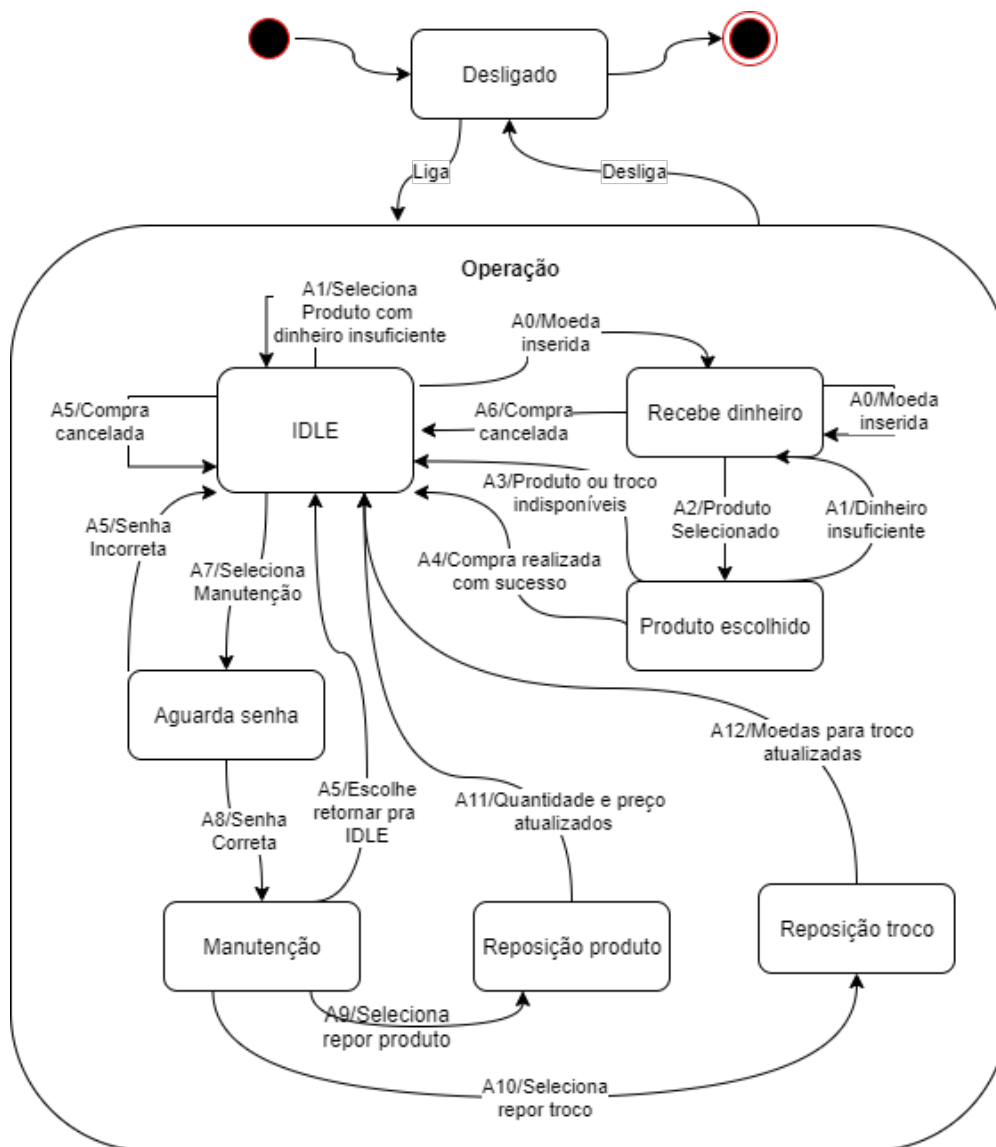


Figura 2: Máquina de Estados.

As ações que são executadas são:

- A0 - Soma e exibe dinheiro;
- A1 - Exibe mensagem dizendo que dinheiro inserido é insuficiente;
- A2 - Verifica se há produto, dinheiro e troco suficientes;
- A3 - Devolve o dinheiro e exibe o que não era suficiente;
- A4 - Atualiza as DBs e libera troco e produto, e volta para opções iniciais do IDLE;

- A5 - Printa display inicial;
- A6 - Devolve dinheiro e printa display inicial;
- A7 - Pedir a senha;
- A8 - Pergunta o que deseja fazer;
- A9 - Confirma que foi selecionado produto;
- A10 - Confirma que foi selecionado troco;
- A11 - Pedir preço, quantidade e localização para atualizar a DB de produto;
- A12 - Pedir quantidade de moedas adicionadas e atualiza DB de troco.

1.3 Diagrama de Componentes

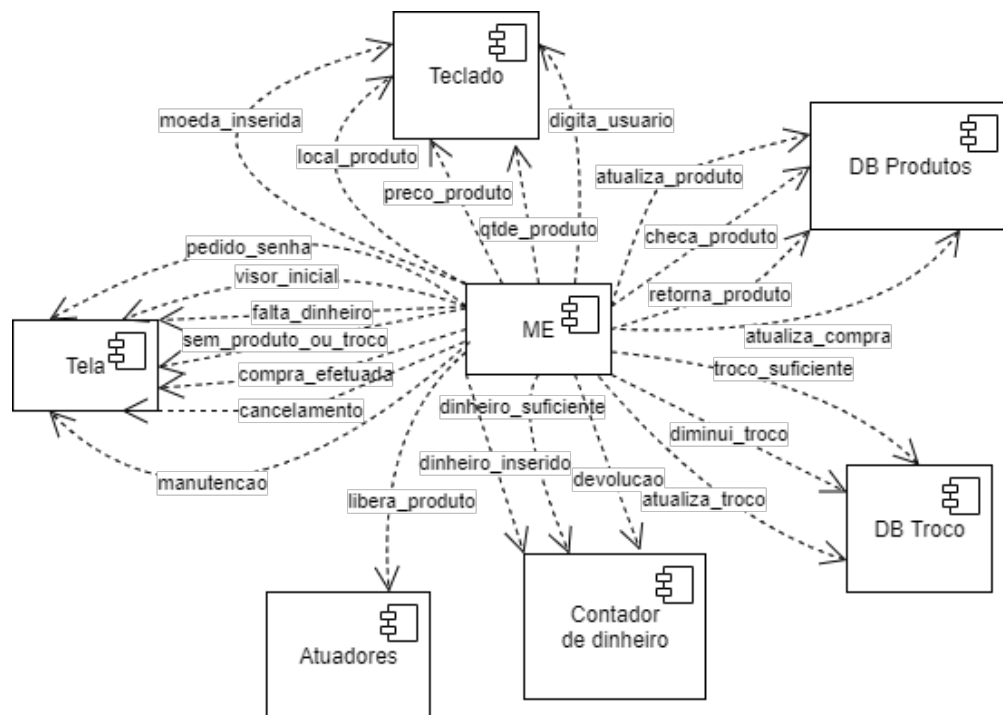


Figura 3: Diagrama de Componentes.

1.4 Diagramas de Sequência

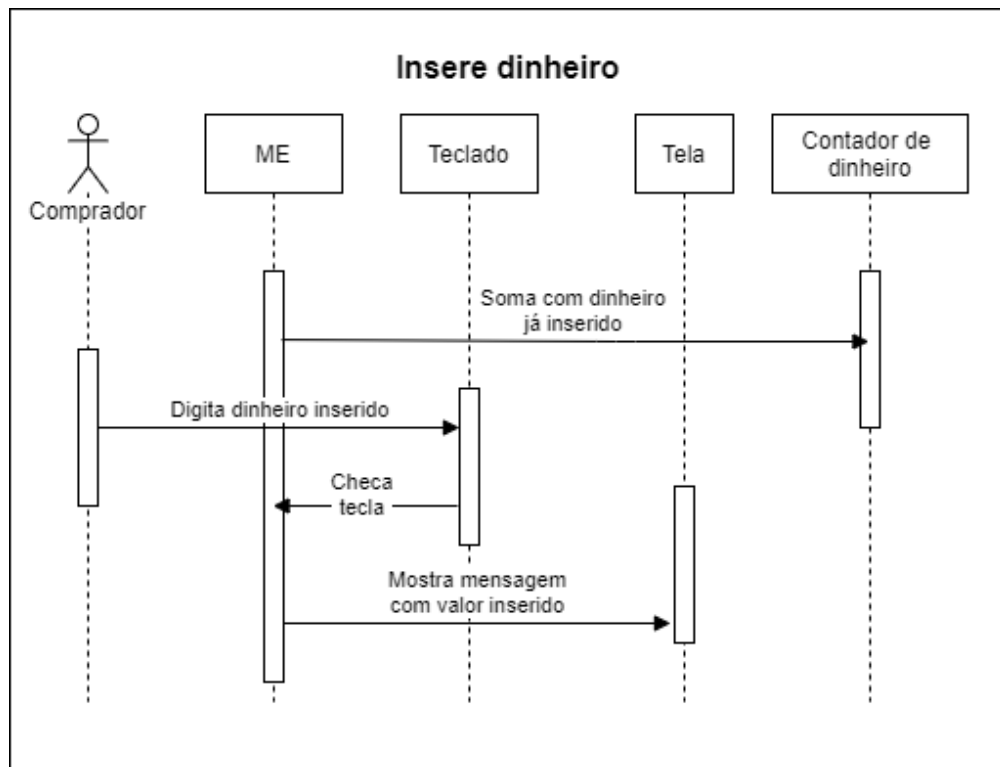


Figura 4: Diagrama de Sequência - Inserir dinheiro.

Seleciona produto

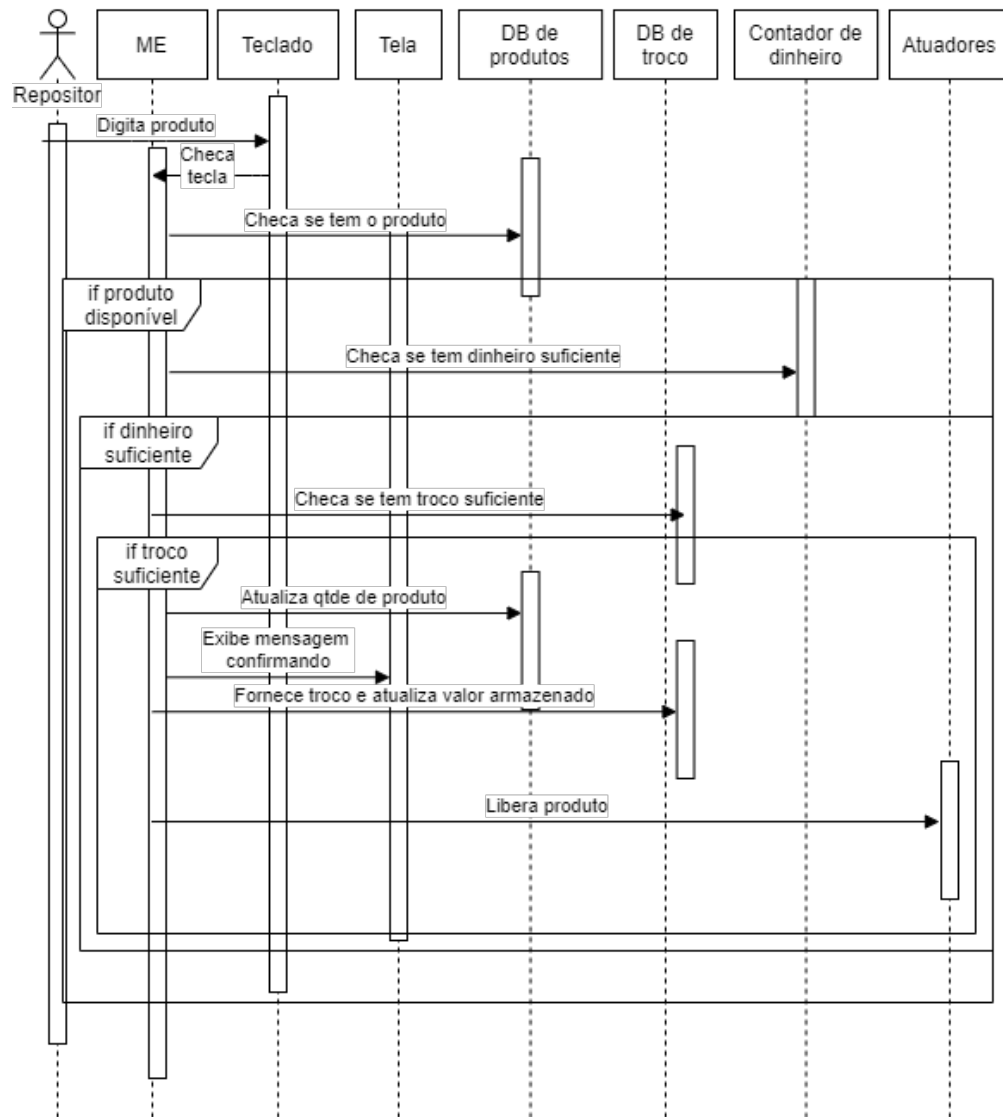


Figura 5: Diagrama de Sequência - Seleciona produto.

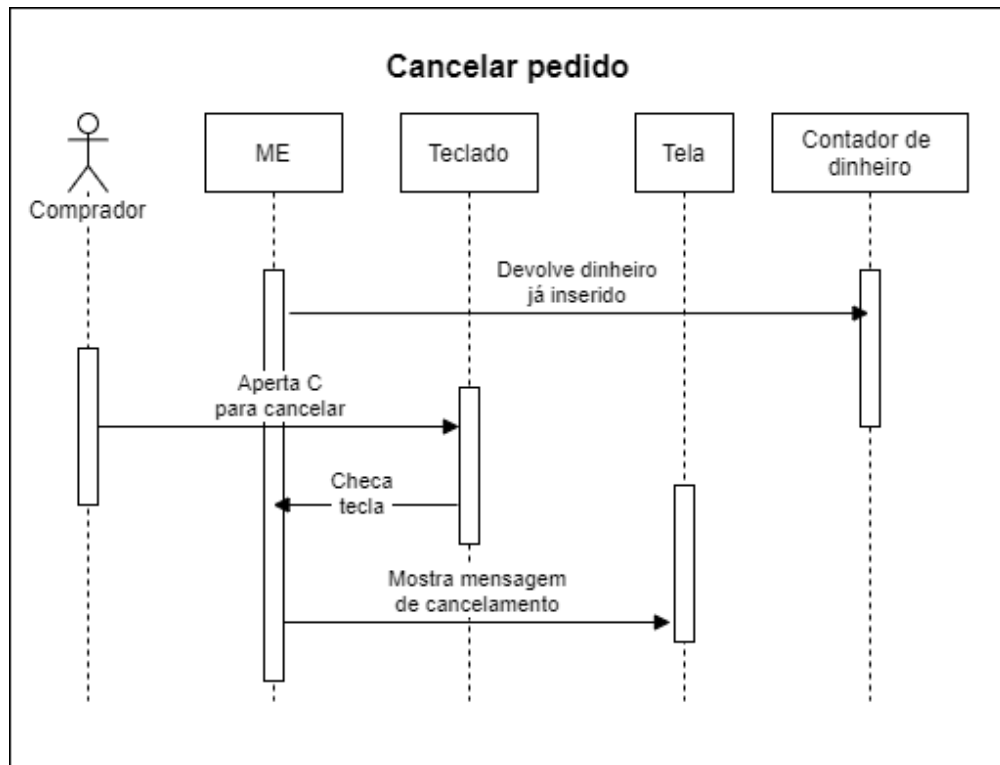


Figura 6: Diagrama de Sequência - Cancelamento.

Cadastra produto ou troco

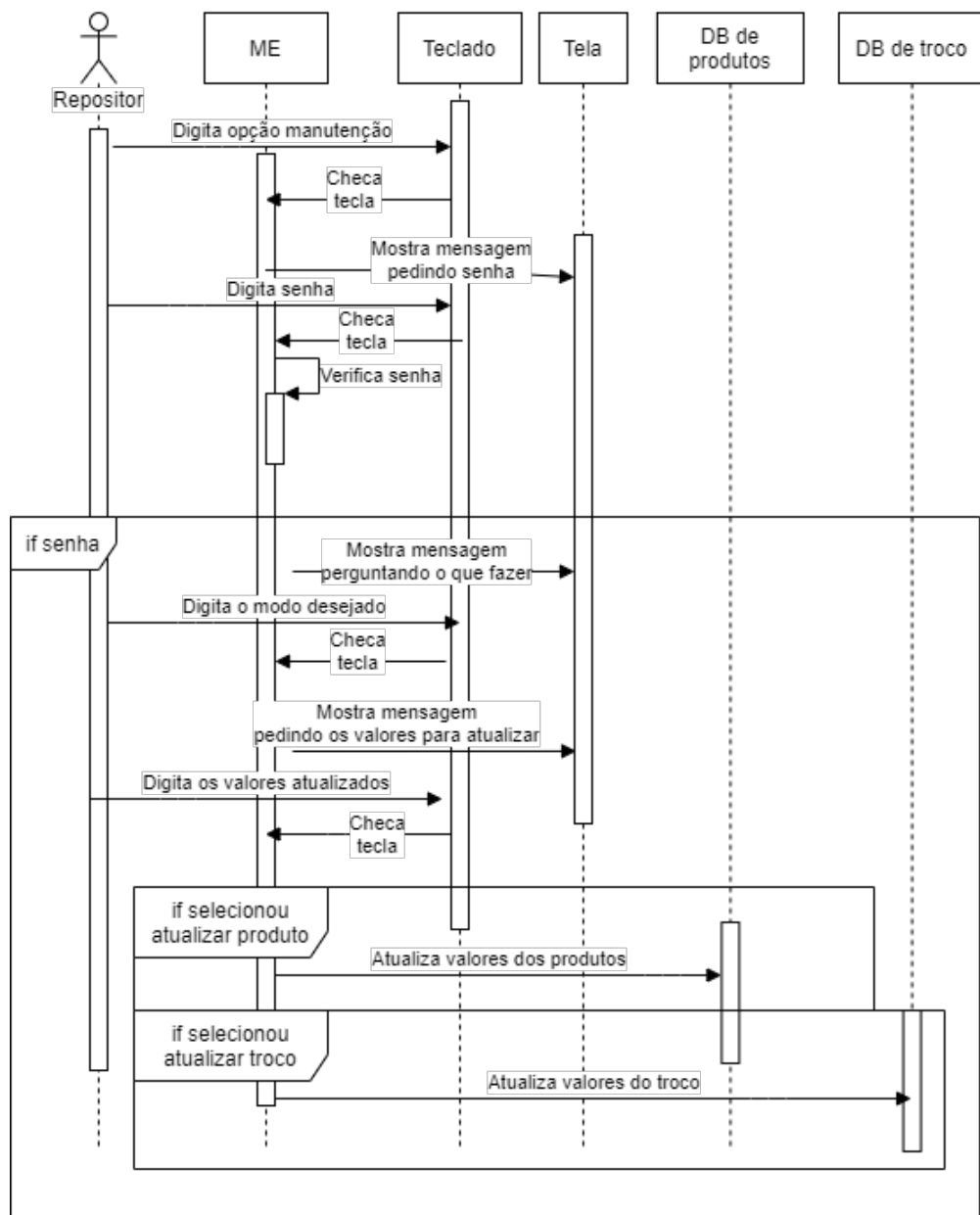


Figura 7: Diagrama de Sequência - Cadastra troco ou produto.

2 Código

2.1 main.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdbool.h>

```

```

4 #include "atuadores.h"
5 #include "dbtroco.h"
6 #include "definicoes_sistema.h"
7 #include "tela.h"
8 #include "gerenciador_de_modos.h"
9 #include "contadordinheiro.h"
10 #include "dbproduto.h"
11 #include "teclado.h"
12
13 //Vending Machine
14 //Rafael Sobral Augusto 10337193 05/05/2020
15
16 //Variaveis que v o ser usadas no c digo todo
17 int acao_transicao[NUM_ESTADOS][NUM_EVENTOS];
18 int estado_seguinte_transicao[NUM_ESTADOS][NUM_EVENTOS];
19 int estado_atual;
20 int evento_atual;
21 int acao_atual;
22 char* desejo;
23 int dinheiro_suficiente;
24 int produto_disponivel;
25 int produto_atual;
26 int troco_retornar;
27
28 //Funcao inicioVM
29 //Como se ligasse a Vending Machine para funcionar
30 void inicioVM(){
31     inicioME();
32 }
33
34 //Funcao inicioME
35 //Atualiza as matrizes acao_transicao e estado_seguinte_transicao
36 //Para tal, segue a m quina de estados definida na teoria
37 //Matrizes indicam para onde ir e qual a o executar para
38 //determinado estado e evento realizado
39 void inicioME(){

```



```

39     int i;
40     int j;
41
42     i=0;
43     while (i<NUM_ESTADOS){
44         j=0;
45         while (j<NUM_EVENTOS){
46             acao_transicao[i][j]=SEM_ACAO_NEM_REACAO;
47             estado_seguinte_transicao[i][j]=i;
48             j+=1;
49         }
50         i+=1;
51     }
52
53     //Eventos no Idle
54     estado_seguinte_transicao[IDLE][MOEDA_INSERIDA]=RECEBE_DINHEIRO;
55     acao_transicao[IDLE][MOEDA_INSERIDA]=A0;
56
57     estado_seguinte_transicao[IDLE][SELECIONA_PRODUTO]=IDLE;
58     acao_transicao[IDLE][SELECIONA_PRODUTO]=A1;
59
60     estado_seguinte_transicao[IDLE][COMPRA_CANCELADA]=IDLE;
61     acao_transicao[IDLE][COMPRA_CANCELADA]=A5;
62
63     estado_seguinte_transicao[IDLE][MANUTENCAO_ESCOLHIDA]=
64         AGUARDA_SENHA;
65     acao_transicao[IDLE][MANUTENCAO_ESCOLHIDA]=A7;
66
67     //Eventos apos recebe dinheiro
68     estado_seguinte_transicao[RECEBE_DINHEIRO][MOEDA_INSERIDA]=
69         RECEBE_DINHEIRO;
70     acao_transicao[RECEBE_DINHEIRO][MOEDA_INSERIDA]=A0;
71
72     estado_seguinte_transicao[RECEBE_DINHEIRO][SELECIONA_PRODUTO]=
73         PRODUTO_ESCOLHIDO;
74     acao_transicao[RECEBE_DINHEIRO][SELECIONA_PRODUTO]=A2;

```

```

72
73     estado_seguinte_transicao[RECEBE_DINHEIRO][COMPRA_CANCELADA]=
        IDLE;
74     acao_transicao[RECEBE_DINHEIRO][COMPRA_CANCELADA]=A6;
75
76     //Eventos no Produto Selecionado
77     estado_seguinte_transicao[PRODUTO_ESCOLHIDO][
        SEM_PRODUTO_OU_TROCO]=IDLE;
78     acao_transicao[PRODUTO_ESCOLHIDO][SEM_PRODUTO_OU_TROCO]=A3;
79
80     estado_seguinte_transicao[PRODUTO_ESCOLHIDO][FALTA_DINHEIRO]=
        RECEBE_DINHEIRO;
81     acao_transicao[PRODUTO_ESCOLHIDO][FALTA_DINHEIRO]=A1;
82
83     estado_seguinte_transicao[PRODUTO_ESCOLHIDO][COMPRA_APROVADA]=
        IDLE;
84     acao_transicao[PRODUTO_ESCOLHIDO][COMPRA_APROVADA]=A4;
85
86     //Eventos no Aguarda Senha
87     estado_seguinte_transicao[AGUARDA_SENHA][SENHA_INCORRETA]=IDLE;
88     acao_transicao[AGUARDA_SENHA][SENHA_INCORRETA]=A5;
89
90     estado_seguinte_transicao[AGUARDA_SENHA][SENHA_CORRETA]=
        MANUTENCAO;
91     acao_transicao[AGUARDA_SENHA][SENHA_CORRETA]=A8;
92
93     //Eventos no Idle Manutencao
94     estado_seguinte_transicao[MANUTENCAO][RETORNA_PARA_IDLE]=IDLE;
95     acao_transicao[MANUTENCAO][RETORNA_PARA_IDLE]=A5;
96
97     estado_seguinte_transicao[MANUTENCAO][SELECIONA_ATUALIZA_PRODUTO
        ]=REPOSICAO_PRODUTO;
98     acao_transicao[MANUTENCAO][SELECIONA_ATUALIZA_PRODUTO]=A9;
99
100    estado_seguinte_transicao[MANUTENCAO][SELECIONA_ATUALIZA_TROCO]=
        REPOSICAO_TROCO;

```

```

101     acao_transicao[MANUTENCAO][SELECIONA_ATUALIZA_TROCO]=A10;
102
103     //Eventos no Reposicao produto
104     estado_seguinte_transicao[REPOSICAO_PRODUTO][PRODUTO_ATUALIZADO
105         ]=IDLE;
106     acao_transicao[REPOSICAO_PRODUTO][PRODUTO_ATUALIZADO]=A11;
107
108     //Eventos no Reposicao troco
109     estado_seguinte_transicao[REPOSICAO_TROCO][TROCO_ATUALIZADO]=
110         IDLE;
111     acao_transicao[REPOSICAO_TROCO][TROCO_ATUALIZADO]=A12;
112 }
113
114 //Funcao execucaoAcao
115 // o que propriamente faz a Vending Machine funcionar
116 //Executa acoes nas transicoes entre estados
117 //Input: numero da acao que precisa ser executada
118 int execucaoAcao(int numAcao){
119     int evento_interno_a_fazer=SEM_EVENTOS_FIQUE_EM_CASA;
120     int dinheiro_inserido;
121
122     if (numAcao==SEM_ACAO_NEM_REACAO){
123         return evento_interno_a_fazer;
124     }
125
126     if (numAcao==A0){
127         //Estado S0 e Evento S0, executar soma e exibir o dinheiro
128         if ((desejo[1]=='1')){
129             if (desejo[3]=='0'){
130                 dinheiro_inserido=100;
131             }
132             else{
133                 dinheiro_inserido=10;
134             }
135         }
136         if (desejo[1]=='5'){

```

```

135         dinheiro_inserido=50;
136     }
137     if (desejo[1]=='2'){
138         dinheiro_inserido=25;
139     }
140     ctd_dinheiro_inserido(dinheiro_inserido);
141     tla_visor_inicial();
142 }
143 if (numAcao==A1){
144     tla_falta_dinheiro();
145     tla_visor_inicial();
146 }
147 if (numAcao==A2){
148     produto_atual=dbp_retorna_produto(desejo);
149     produto_disponivel=dbp_checa_produto(produto_atual);
150     if (produto_disponivel==1){
151         dinheiro_suficiente=ctd_dinheiro_suficiente(
152             produto_atual);
153         if (dinheiro_suficiente==1){
154             troco_retornar=dbt_troco_suficiente(dinheiro,
155                 produto_atual);
156             if (troco_retornar<0){
157                 evento_interno_a_fazer=SEM_PRODUTO_OU_TROCO;
158                 troco_retornar=0;
159             }
160             else{
161                 evento_interno_a_fazer=COMPRA_APROVADA;
162             }
163         }
164         else{
165             evento_interno_a_fazer=FALTA_DINHEIRO;
166         }
167     }
168     else{
169         evento_interno_a_fazer=SEM_PRODUTO_OU_TROCO;
170     }

```

```

169     }
170     if (numAcao==A3){
171         //Devolver dinheiro porque faltou o produto ou troco
172         tla_sem_produto_ou_troco(produto_disponivel);
173         ctd_devolucao();
174         tla_visor_inicial();
175     }
176     if (numAcao==A4){
177         //Tudo certo, compra efetuada
178         dbp_atualiza_compra(produto_atual);
179         atd_libera_produto();
180         tla_compra_efetuada(troco_retornar);
181         dbt_diminui_troco();
182         dinheiro=0;
183         troco_retornar=0;
184         tla_visor_inicial();
185     }
186     if (numAcao==A5){
187         tla_visor_inicial();
188     }
189     if (numAcao==A6){
190         tla_cancelamento();
191         ctd_devolucao();
192         tla_visor_inicial();
193     }
194     if (numAcao==A7){
195         tla_pedido_senha();
196     }
197     if (numAcao==A8){
198         tla_manutencao();
199     }
200     if (numAcao==A9){
201         tla_confirma_manutencao(1);
202         evento_interno_a_fazer=PRODUTO_ATUALIZADO;
203     }
204     if (numAcao==A10){

```

```

205         tla_confirma_manutencao(2);
206         evento_interno_a_fazer=TROCO_ATUALIZADO;
207     }
208     if (numAcao==A11){
209         int qtde;
210         int preco;
211         int local;
212         local=tec_local_produto();
213         qtde=tec_qtde_produto();
214         preco=tec_preco_produto();
215         dbp_atualiza_produto(local,qtde,preco);
216         tla_visor_inicial();
217     }
218     if (numAcao==A12){
219         int moedas[4];
220         int i=0;
221         while (i<=3){
222             moedas[i]=tec_moeda_inserida(i);
223             i+=1;
224         }
225         dbt_atualiza_troco(moedas);
226         tla_visor_inicial();
227     }
228     return evento_interno_a_fazer;
229 }
230
231 //Funcao obtemEvento
232 //Interpreta qual deve ser o proximo evento de acordo com o estado e
    o que aconteceu
233 //Depende exclusivamente da acao do usu rio
234 //Output: numero do evento que tem que ser realizado
235 int obtemEvento(){
236     int evento_obtido=SEM_EVENTOS_FIQUE_EM_CASA;
237     desejo=tec_digita_usuario();
238     if ((estado_atual==IDLE || estado_atual==RECEBE_DINHEIRO)&& (
        desejo[0]=='x')){

```

```

239         //Estado S0 e Evento E0
240         return MOEDA_INSERTIDA;
241     }
242     else if ((estado_atual==IDLE || estado_atual==RECEBE_DINHEIRO)
243             && desejo[0]=='p'){
244         return SELECIONA_PRODUTO;
245     }
246     else if ((estado_atual==IDLE || estado_atual==RECEBE_DINHEIRO)
247             && desejo[0]=='C'){
248         return COMPRA_CANCELADA;
249     }
250     else if (estado_atual==IDLE && desejo[0]=='M'){
251         return MANUTENCAO_ESCOLHIDA;
252     }
253     else if (estado_atual==AGUARDA_SENHA){
254         if (strcmp(desejo,"covid19")==0){
255             return SENHA_CORRETA;
256         }
257         return SENHA_INCORRETA;
258     }
259     else if (estado_atual==MANUTENCAO){
260         if (desejo[0]=='P'){
261             return SELECIONA_ATUALIZA_PRODUTO;
262         }
263         else if (desejo[0]=='T'){
264             return SELECIONA_ATUALIZA_TROCO;
265         }
266         else{
267             return RETORNA_PARA_IDLE;
268         }
269     }
270     else if (estado_atual==REPOSICAO_PRODUTO){
271         return PRODUTO_ATUALIZADO;
272     }
273     return evento_obtido;
274 }

```

```

273
274 //Funcao obtemAcao
275 //Basicamente, pega todo o trabalho ja feito pela obtemEvento e com
        isso, decide a acao a realizar
276 //Baseia-se na matriz inicial montada em inicioME
277 //Inputs: Estado e eventos atuais
278 //Output: Acao a ser realizada agora
279 int obtemAcao(int estado_atual, int evento_atual){
280     return acao_transicao[estado_atual][evento_atual];
281 }
282
283 //Funcao obtemProximoEstado
284 //Basicamente, tambem pega o trabalho realizado pela obtemEvento e
        fala qual eh o estado pra onde vai
285 //Baseia-se puramente na matriz montada em inicioME
286 //Inputs: Estado e eventos atuais
287 //Output: Proximo estado
288 int obtemProximoEstado(int estado_atual, int evento_atual){
289     return estado_seguinte_transicao[estado_atual][evento_atual];
290 }
291
292 //Funcao Main
293 //Responsavel por inicializar a maquina e o loop infinito da VM
294 int main(){
295     int evento_interno;
296
297     evento_interno=SEM_EVENTOS_FIQUE_EM_CASA;
298     inicioVM();
299     printf("IT'S ALIVEEEE!! By: Sobral. \n");
300
301     estado_atual=IDLE;
302
303     //Realiza o print do display inicial
304     tlavisor_inicial();
305
306     while (1){

```



```

307         if (evento_interno==SEM_EVENTOS_FIQUE_EM_CASA){
308             //Caso nao tenham eventos internos pra realizar, pede
                 input do usu rio
309             evento_atual=obtemEvento();
310         }
311         else {
312             evento_atual=evento_interno;
313         }
314         if (evento_atual!=SEM_EVENTOS_FIQUE_EM_CASA) {
315             acao_atual=obtemAcao(estado_atual, evento_atual);
316             estado_atual=obtemProximoEstado(estado_atual,
                 evento_atual);
317             evento_interno=execucaoAcao(acao_atual);
318         }
319     }
320 }

```

2.2 definicoes_{sistema.h}

```

1  #ifndef DEFINICOES_SISTEMA_H_INCLUDED
2  #define DEFINICOES_SISTEMA_H_INCLUDED
3
4  #define NUM_ESTADOS 7
5  #define NUM_EVENTOS 13
6
7  // ESTADOS
8  #define IDLE 0
9  #define RECEBE_DINHEIRO 1
10 #define PRODUTO_ESCOLHIDO 2
11 #define AGUARDA_SENHA 3
12 #define MANUTENCAO 4
13 #define REPOSICAO_PRODUTO 5
14 #define REPOSICAO_TROCO 6
15
16 // EVENTOS
17 #define SEM_EVENTOS_FIQUE_EM_CASA -1
18 #define MOEDA_INSERTIDA 0

```

```

19 #define SELECIONA_PRODUTO 1
20 #define SEM_PRODUTO_OU_TROCO 2
21 #define FALTA_DINHEIRO 3
22 #define COMPRA_APROVADA 4
23 #define COMPRA_CANCELADA 5
24 #define MANUTENCAO_ESCOLHIDA 6
25 #define SENHA_INCORRETA 7
26 #define SENHA_CORRETA 8
27 #define RETORNA_PARA_IDLE 9
28 #define SELECIONA_ATUALIZA_PRODUTO 10
29 #define SELECIONA_ATUALIZA_TROCO 11
30 #define PRODUTO_ATUALIZADO 12
31 #define TROCO_ATUALIZADO 13
32
33 // ACOES
34 #define SEM_ACAO_NEM_REACAO -1
35 #define A0 0 //Soma dinheiro e exhibe
36 #define A1 1 //Exibe mensagem pedindo dinheiro
37 #define A2 2 //Verifica produto dinheiro e troco
38 #define A3 3 //Devolve dinheiro e exhibe problema (produto ou troco)
39 #define A4 4 //Atualiza DB de troco/produto e os libera
40 #define A5 5 //Printa o inicio no Idle
41 #define A6 6 //Devolve dinheiro e printa inicio do Idle
42 #define A7 7 //Pede a senha para o usuario
43 #define A8 8 //Pergunta o que deseja fazer
44 #define A9 9 //Confirma que selecionou repor produto
45 #define A10 10 //Confirma que selecionou repor troco
46 #define A11 11 //Pede preco, qtde e local e atualiza DB de produtos
    e printa valores atuais
47 #define A12 12 //Pede valores das moedas, atualiza DB de troco e
    printa valores atuais
48
49 #endif

```

2.3 Demais headers

2.3.1 atuadores.h

```
1 #ifndef ATUADORES_H_INCLUDED
2 #define ATUADORES_H_INCLUDED
3
4 extern void atd_libera_produto();
5
6 #endif
```

2.3.2 contadordinheiro.h

```
1 #ifndef CONTADORDINHEIRO_H_INCLUDED
2 #define CONTADORDINHEIRO_H_INCLUDED
3
4 extern int dinheiro;
5
6 extern void ctd_dinheiro_inserido(int dinheiro_inserido);
7
8 extern int ctd_dinheiro_suficiente(int produto_atual);
9
10 extern void ctd_devolucao();
11
12 #endif
```

2.3.3 dbproduto.h

```
1 #ifndef DBPRODUTO_H_INCLUDED
2 #define DBPRODUTO_H_INCLUDED
3
4 extern char Produtos_nome[10][25];
5 extern int Produtos_preco[10];
6 extern int Produtos_qtde[10];
7
8 extern int dbp_checa_produto(int produto_atual);
9
10 extern int dbp_retorna_produto(char* desejo);
11
```

```

12 extern void dbp_atualiza_compra(int produto_atual);
13
14 extern void dbp_atualiza_produto(int local, int qtde, int preco);
15
16 #endif

```

2.3.4 dbtroco.h

```

1 #ifndef DBTROCO_H_INCLUDED
2 #define DBTROCO_H_INCLUDED
3
4 extern int troco[4];
5
6 extern int dbt_troco_suficiente(int dinheiro, int produto_atual);
7
8 extern void dbt_diminui_troco();
9
10 extern void dbt_atualiza_troco(int moedas[4]);
11
12 #endif

```

2.3.5 teclado.h

```

1 #ifndef TECLADO_H_INCLUDED
2 #define TECLADO_H_INCLUDED
3
4 extern char* tec_digita_usuario();
5
6 extern int tec_qtde_produto();
7
8 extern int tec_preco_produto();
9
10 extern int tec_local_produto();
11
12 extern int tec_moeda_inserida(int i);
13
14 #endif

```

2.3.6 tela.h

```
1 #ifndef TELA_H_INCLUDED
2 #define TELA_H_INCLUDED
3
4 extern void tla_visor_inicial();
5
6 extern void tla_falta_dinheiro();
7
8 extern void tla_sem_produto_ou_troco(int produto_disponivel);
9
10 extern void tla_compra_efetuada(int troco_retornar);
11
12 extern void tla_cancelamento();
13
14 extern void tla_pedido_senha();
15
16 extern void tla_manutencao();
17
18 extern void tla_confirma_manutencao(int decisao);
19
20 #endif
```

2.4 Sources

2.4.1 atuadores.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 extern void atd_libera_produto(){
6     //Finge que fez algo, pq nao tem um atuador em C
7     //Nao deveria ter uns prints, todo print deveria estar no
8     //componente tela, soh pus pra simular mais a experiencia do
9     //comprador
10
11     int trava;
12     printf("\nGira gira gira gira... \n");
```

```

10     trava=rand();
11     if (trava<=25){
12         printf("Pois eh, parece que travou. Uma pena, da umas
           batidinhas ai!\n");
13     }
14 }

```

2.4.2 contadordinheiro.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <stdbool.h>
5  #include "dbproduto.h"
6  #include "definicoes_sistema.h"
7  #include "contadordinheiro.h"
8
9  //V riavel que armazena dinheiro inserido pelo comprador
10 extern int dinheiro=0;
11
12 //Incrementa o dinheiro inserido conforme a moeda selecionada
13 extern void ctd_dinheiro_inserido(int dinheiro_inserido){
14     dinheiro+=dinheiro_inserido;
15 }
16
17 //Checa se o dinheiro inserido      suficiente para o produto
    selecionado
18 extern int ctd_dinheiro_suficiente(int produto_atual){
19     if (Produtos_preco[produto_atual]<=dinheiro){
20         return 1;
21     }
22     return 0;
23 }
24
25 // "Devolve" o dinheiro do comprador caso cancele a compra ou n o
    tenha troco/produto
26 extern void ctd_devolucao(){

```

```

27     dinheiro=0;
28 }

```

2.4.3 dbproduto.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <stdbool.h>
5  #include "definicoes_sistema.h"
6  #include "dbproduto.h"
7
8  //Listas responsaveis por ter o preco, a quantidade e o nome dos
   produtos
9  extern int Produtos_preco[10]={350, 300, 250, 200, 300, 300, 350,
   400, 0, 0};
10 extern int Produtos_qtde[10]={7,3,9,2,4,0,12,3,3,8};
11 extern char Produtos_nome[10][25]={"Alcool em gel","Mascara","
   Isolamento","Quarentena","Distanciamento","Respirador","Leito","
   Hospital","Meets","10naVM"};
12
13 //Checa se tem produto para suprir o pedido do usuario
14 extern int dbp_checa_produto(int produto_atual){
15     if (Produtos_qtde[produto_atual]>0){
16         return 1;
17     }
18     return 0;
19 }
20
21 //Retorna o produto que o usuario deseja de acordo com o input
22 //Tive que fazer uma adapta o pois stoi e atoi n o funcionam no
   meu pc
23 extern int dbp_retorna_produto(char* desejo){
24     if (desejo[1]=='1'){
25         if (desejo[2]=='0'){
26             return 9;
27         }

```

```

28         return 0;
29     }
30     if (desejo[1]=='2'){
31         return 1;
32     }
33     if (desejo[1]=='3'){
34         return 2;
35     }
36     if (desejo[1]=='4'){
37         return 3;
38     }
39     if (desejo[1]=='5'){
40         return 4;
41     }
42     if (desejo[1]=='6'){
43         return 5;
44     }
45     if (desejo[1]=='7'){
46         return 6;
47     }
48     if (desejo[1]=='8'){
49         return 7;
50     }
51     if (desejo[1]=='9'){
52         return 8;
53     }
54 }
55
56 //Atualiza a DB do produto ap s a compra ser confirmada
57 extern void dbp_atualiza_compra(int produto_atual){
58     Produtos_qtde[produto_atual]-=1;
59 }
60
61 //Atualiza a DB do produto apos reposi o na manuten o
62 extern void dbp_atualiza_produto(int local, int qtde, int preco){
63     Produtos_qtde[local]+=qtde;

```



```

64     Produtos_preco[local]=preco;
65 }

```

2.4.4 dbtroco.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <stdbool.h>
5  #include "definicoes_sistema.h"
6  #include "contadordinheiro.h"
7  #include "dbproduto.h"
8  #include "dbtroco.h"
9
10 //Variaveis para conter quanto troco tem e calcular quanto troco
    precisa sem alterar a DB do troco
11 extern int troco[4]={10,8,15,7};
12 extern int usados[4]={0,0,0,0};
13 int trocoretornar;
14
15 //Calcula qual o troco necessario e se tem esse valor
16 extern int dbt_troco_suficiente(int dinheiro,int produto_atual){
17     int trocoadar=dinheiro-Produtos_preco[produto_atual];
18     trocoretornar=trocoadar;
19
20     while (trocoadar>0){
21         if (trocoadar>=10){
22             if (trocoadar%10!=0){
23                 if (troco[1]-usados[1]>0){
24                     trocoadar -=25;
25                     usados[1] +=1;
26                 }
27             }
28             else if (trocoadar>=100){
29                 if (troco[3]-usados[3]>0){
30                     trocoadar -=100;
31                     usados[3] +=1;

```

```

32         }
33         else if (troco[2]-usados[2]>1){
34             trocoadar -=100;
35             usados[2] +=1;
36         }
37         else if (troco[1]-usados[1]>3){
38             trocoadar -=100;
39             usados[1] +=4;
40         }
41         else if (troco[0]-usados[0]>9){
42             trocoadar -=100;
43             troco[0] -=10;
44             usados[0] +=10;
45         }
46         else{
47             usados[0]=0;
48             usados[1]=0;
49             usados[2]=0;
50             usados[3]=0;
51             return -1;
52         }
53     }
54     else if (trocoadar >=50){
55         if (troco[2]-usados[2]>0){
56             trocoadar -=50;
57             usados[2] +=1;
58         }
59         else if (troco[1]-usados[1]>1){
60             trocoadar -=50;
61             usados[1] +=1;
62         }
63         else if (troco[0]-usados[0]>4){
64             trocoadar -=50;
65             usados[0] +=5;
66         }
67         else{

```

```

68         usados [0]=0;
69         usados [1]=0;
70         usados [2]=0;
71         usados [3]=0;
72         return -1;
73     }
74 }
75 else{
76     if (troco [0]-usados [0]>0){
77         usados [0]+=1;
78         trocoadar -=10;
79     }
80     else{
81         usados [0]=0;
82         usados [1]=0;
83         usados [2]=0;
84         usados [3]=0;
85         return -1;
86     }
87 }
88 }
89 else{
90     usados [0]=0;
91     usados [1]=0;
92     usados [2]=0;
93     usados [3]=0;
94     return -1;
95 }
96 }
97 return trocoretornar;
98 }
99
100 //Atualiza a DB do troco apos compra confirmada
101 extern void dbt_diminui_troco(){
102     troco [0] -=usados [0];
103     troco [1] -=usados [1];

```

```

104     troco[2]-=usados[2];
105     troco[3]-=usados[3];
106     usados[0]=0;
107     usados[1]=0;
108     usados[2]=0;
109     usados[3]=0;
110 }
111
112 //Atualiza o troco apos Manuten o
113 extern void dbt_atualiza_troco(int moedas[4]){
114     int i=0;
115     while (i<=3){
116         troco[i]+=moedas[i
117         i+=1;
118     }
119 }

```

2.4.5 teclado.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include "definicoes_sistema.h"
5  #include "teclado.h"
6
7  //Char para pegar o valor digitado
8  char numerodigitado[10];
9  int qtde;
10 int preco;
11 int local;
12
13 // o obter_tecclas que recebe o que foi digitado pelo usu rio
14 extern char* tec_digita_usuario(){
15     printf("Digite aqui: ");
16     scanf("%s", numerodigitado);
17     return numerodigitado;
18 }

```

```

19
20 //Serve unicamente para a Manutenção, para pegar o local em que
    deseja inserir o produto
21 extern int tec_local_produto(){
22     printf("Digite o local do produto que vai atualizar: ");
23     scanf("%d",&local);
24     return local-1;
25 }
26
27 //Serve unicamente para a Manutenção, para pegar a qtde que deseja
    inserir do produto
28 extern int tec_qtde_produto(){
29     printf("Digite aqui a quantidade adicionada: ");
30     scanf("%d",&qtde);
31     return qtde;
32 }
33
34 //Serve unicamente para a Manutenção, para pegar o preço que
    deseja inserir do produto
35 extern int tec_preco_produto(){
36     printf("Digite aqui o preço do produto (em centavos): ");
37     scanf("%d",&preco);
38     return preco;
39 }
40
41 //Serve unicamente para a Manutenção, para pegar quantas moedas de
    troco ele inseriu
42 extern int tec_moeda_inserida(int i){
43     int moeda;
44     if (i==0){
45         moeda=10;
46     }
47     else if (i==1){
48         moeda=25;
49     }
50     else if (i==2){

```

```

51     moeda=50;
52 }
53 else{
54     moeda=100;
55 }
56 printf("Digite aqui quantas de %d centavos deseja inserir: ",
57        moeda);
58 scanf("%d",&moeda);
59 return moeda;
60 }

```

2.4.6 tela.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include "definicoes_sistema.h"
5  #include "contadordinheiro.h"
6  #include "dbproduto.h"
7  #include "tela.h"
8
9  //Visor padrao da VM, aparecer sempre que voltar para o IDLE
10 extern void tla_visor_inicial(){
11     printf("\nVending Machine da Quarentena! \n");
12     printf("Escolha dentre os seguintes produtos: \n");
13
14     int i;
15     i=0;
16     float precofloat;
17     while (i<10){
18         precofloat=Produtos_preco[i]/100.0;
19         printf("%d - %s : R$ %0.2f \n", i+1, Produtos_nome[i],
20                precofloat);
21         i+=1;
22     }
23     float dinheirofloat;
24     dinheirofloat=dinheiro/100.0;

```

```

24     printf("\nPara inserir dinheiro, digite x+o valor da sua moeda
        em centavos! \n");
25     printf("Por exemplo, se inserir 100 centavos, digitar x100 \n");
26     printf("Para selecionar um produto, digite p+o local do produto
        \n");
27     printf("Por exemplo, se quiser o produto 5, digitar p5 \n");
28     printf("Para entrar na Manutenção, digite M \n");
29     printf("Você inseriu: R$%0.2f \n",dinheirofloat);
30     printf("Para cancelar a operação, digite C \n");
31 }
32
33 //Mensagem avisando que não tem dinheiro suficiente para certo
    produto
34 extern void tla_falta_dinheiro(){
35     printf("\nINSIRA DINHEIRO SUFICIENTE PARA ESSE PRODUTO!! \n \n")
        ;
36 }
37
38 //Mensagem avisando erro em alguma das etapas da compra
39 extern void tla_sem_produto_ou_troco(int produto_disponivel){
40     if (produto_disponivel==0){
41         printf("\nInfelizmente, esse produto está em falta! Pegue
            seu dinheiro de volta ao lado e volte para casa em
            segurança! \n");
42     }
43     else{
44         printf("\nInfelizmente, não temos troco suficiente para o
            valor inserido! Pegue seu dinheiro de volta ao lado! \n")
            ;
45     }
46 }
47
48 //Mensagem avisando que a compra foi efetuada direitinho
49 extern void tla_compra_efetuada(int troco_retornar){
50     printf("\nSua compra foi efetuada com sucesso! \n");
51     printf("Retire seu produto e seu troco de R$%0.2f! \n \n",

```

```

        troco_retornar/100.0);
52 }
53
54 //Mensagem confirmando o cancelamento
55 extern void tla_cancelamento(){
56     printf("\nInfelizmente, sua compra foi CANCELADAH! Pegue seu
        dinheiro de volta ao lado! \n");
57 }
58
59 //Mensagem pedindo a senha na manuten o
60 extern void tla_pedido_senha(){
61     printf("\nPor favor, digite a senha correta abaixo para entrar
        no modo de Manutencao! \n");
62 }
63
64 //Mensagem para quando entrou na manuten o
65 extern void tla_manutencao(){
66     printf("\nBem-vindo ao modo de Manutencao! \n");
67     printf("O que voc deseja fazer? \n");
68     printf("Digite T se quiser atualizar o troco \n");
69     printf("Digite P se quiser atualizar os produtos \n");
70     printf("Digite qualquer outra coisa para determinar o fim e
        voltar para a tela inicial \n");
71 }
72
73 //Mensagem confirmando o que foi selecionado
74 extern void tla_confirma_manutencao(int decisao){
75     if (decisao==1){
76         printf("Voce selecionou atualizar as informacoes dos
            produtos! \n");
77     }
78     else{
79         printf("Voce selecionou atualizar as informacoes do troco! \
            n");
80     }
81 }

```


3 Prints do funcionamento

Abaixo, seguem imagens com os prints do funcionamento da Vending Machine, como solicitado. Tentei seguir todos os fluxos de funcionamento possíveis do programa. Ao fim deles, só reseta pra Idle e printa a mesma mensagem inicial.

3.1 Tela inicial

```
IT'S ALIVEEEE!! By: Sobral.  
  
Vending Machine da Quarentena!  
Escolha dentre os seguintes produtos:  
1 - Alcool em gel : R$ 3.50  
2 - Mascara : R$ 3.00  
3 - Isolamento : R$ 2.50  
4 - Quarentena : R$ 2.00  
5 - Distanciamento : R$ 3.00  
6 - Respirador : R$ 3.00  
7 - Leito : R$ 3.50  
8 - Hospital : R$ 4.00  
9 - Meets : R$ 0.00  
10 - 10naVM : R$ 0.00  
  
Para inserir dinheiro, digite x+o valor da sua moeda em centavos!  
Por exemplo, se inserir 100 centavos, digitar x100  
Para selecionar um produto, digite p+o local do produto  
Por exemplo, se quiser o produto 5, digitar p5  
Para entrar na Manutenção, digite M  
Você já inseriu: R$0.00  
Para cancelar a operacao, digite C  
Digite aqui:
```

Figura 8: Tela Inicial - Vending Machine.

3.2 Tela após inserir dinheiro

Houve diferença na quantidade de dinheiro já inserido na máquina.

```
Vending Machine da Quarentena!  
Escolha dentre os seguintes produtos:  
1 - Alcool em gel : R$ 3.50  
2 - Mascara : R$ 3.00  
3 - Isolamento : R$ 2.50  
4 - Quarentena : R$ 2.00  
5 - Distanciamento : R$ 3.00  
6 - Respirador : R$ 3.00  
7 - Leito : R$ 3.50  
8 - Hospital : R$ 4.00  
9 - Meets : R$ 0.00  
10 - 10naVM : R$ 0.00  
  
Para inserir dinheiro, digite x+o valor da sua moeda em centavos!  
Por exemplo, se inserir 100 centavos, digitar x100  
Para selecionar um produto, digite p+o local do produto  
Por exemplo, se quiser o produto 5, digitar p5  
Para entrar na Manutenção, digite M  
Você já inseriu: R$1.00  
Para cancelar a operacao, digite C  
Para escolher um produto, digite o número correspondente  
Digite aqui: _
```

Figura 9: Tela após inserir dinheiro - Vending Machine.

3.3 Aviso de dinheiro insuficiente

```
INSIRA DINHEIRO SUFICIENTE PARA ESSE PRODUTO!!
```

Figura 10: Dinheiro insuficiente - Vending Machine.

3.4 Aviso de produto indisponível

```
Infelizmente, esse produto está em falta! Pegue seu dinheiro de volta ao lado e volte para casa em segurança!
```

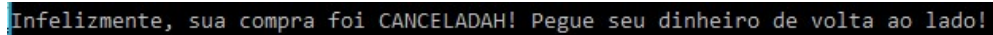
Figura 11: Produto indisponível - Vending Machine.

3.5 Aviso de troco indisponível

```
Infelizmente, não temos troco suficiente para o valor inserido! Pegue seu dinheiro de volta ao lado!
```

Figura 12: Troco indisponível - Vending Machine.

3.6 Aviso de compra cancelada

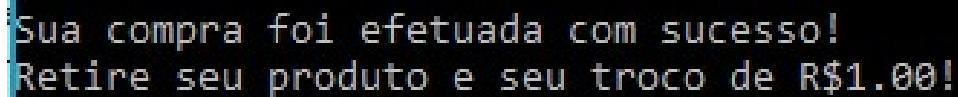


```
Infelizmente, sua compra foi CANCELADA! Pegue seu dinheiro de volta ao lado!
```

Figura 13: Compra cancelada - Vending Machine.

3.7 Aviso de compra concluída

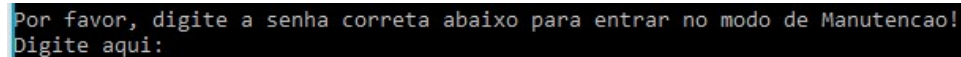
Exibe que a compra foi concluída com o valor do troco calculado.



```
Sua compra foi efetuada com sucesso!  
Retire seu produto e seu troco de R$1.00!
```

Figura 14: Compra concluída - Vending Machine.

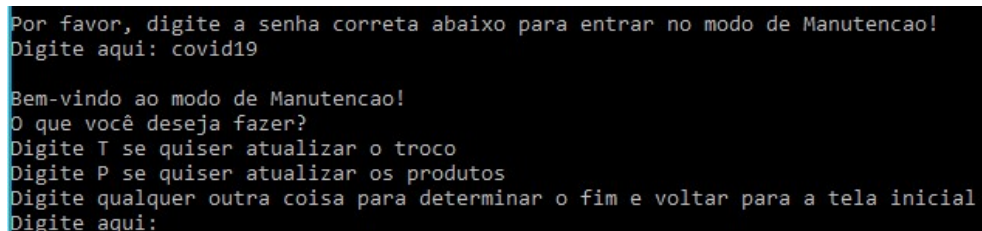
3.8 Mensagem para entrar no modo manutenção



```
Por favor, digite a senha correta abaixo para entrar no modo de Manutencao!  
Digite aqui:
```

Figura 15: Mensagem pedindo a senha - Vending Machine.

3.9 Entrada Modo Manutenção



```
Por favor, digite a senha correta abaixo para entrar no modo de Manutencao!  
Digite aqui: covid19  
  
Bem-vindo ao modo de Manutencao!  
O que você deseja fazer?  
Digite T se quiser atualizar o troco  
Digite P se quiser atualizar os produtos  
Digite qualquer outra coisa para determinar o fim e voltar para a tela inicial  
Digite aqui:
```

Figura 16: Entrada Modo Manutenção - Vending Machine.

3.10 Mensagens para adicionar produto

Mensagens exibidas caso o repositor tenha selecionado adicionar produtos. Não permite mudar o nome unicamente porque essa Vending Machine é temática da quarentena, portanto não seriam mudados os nomes.

```
Voce selecionou atualizar as informacoes dos produtos!  
Digite o local do produto que vai atualizar: 2  
Digite aqui a quantidade adicionada: 10  
Digite aqui o preco do produto (em centavos): 500_
```

Figura 17: Mensagens para adicionar produto - Vending Machine.

3.11 Mensagens para adicionar troco

Mensagens exibidas caso o repositor tenha selecionado adicionar troco.

```
Voce selecionou atualizar as informacoes do troco!  
Digite aqui quantas de 10 centavos deseja inserir: 18_  
Digite aqui quantas de 25 centavos deseja inserir: 29  
Digite aqui quantas de 50 centavos deseja inserir: 13  
Digite aqui quantas de 100 centavos deseja inserir: 1
```

Figura 18: Mensagens para adicionar troco - Vending Machine.

4 Opinião sobre a matéria

No geral, eu achei importante aprender sobre os diagramas UML e o uso de uma estrutura para organizar todo o processo antes de fazer o código. Mas de verdade, teria ajudado muito se eu tivesse visto o objetivo final antes de fazer diagramas sem ter certeza de qual seria o fim direito. Se eu soubesse como tinha que ser a estrutura, me ajudaria muito mais. E honestamente, os diagramas por si só não me ajudaram em nada, foi mais fácil me organizar de outra forma, mas entendo que era só um exemplo.