

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica
PMR3402 - Sistemas Embarcados

Modelagem de uma *Vending Machine*

Luiz Henrique Souza Caldas
11181038

5 de Maio de 2020

1 Introdução

A *Vending Machine* é uma máquina que oferta itens como salgadinhos, biscoitos, doces ou bebidas ao consumidor após o mesmo inserir dinheiro, um cartão de crédito ou um cartão específico para a máquina. Neste trabalho será implementada uma *Vending Machine* com capacidade de armazenar 6 diferentes produtos com até 20 unidades por produto.

O consumidor poderá escolher entre os produtos selecionando o código do produto que deseja. O pagamento deverá ser feito exclusivamente em moedas, em moedas de 10, 25 ou 50 centavos e moedas de 1 real, e o troco será dado em moedas com os mesmos valores. Quando os tubos de moedas para troco estão totalmente carregados, a máquina possui 40 moedas de cada tipo, totalizando 74 reais em troco. Se a máquina não tiver as moedas necessárias para o troco, as moedas inseridas são devolvidas. Após completado o pagamento, um arame helicoidal referente ao produto selecionado é acionado por um motor, liberando o produto no compartimento de retirada.

Para efetuar essas atividades, a máquina deverá conter os seguintes componentes: teclado numérico com números de 0 a 9, botão de cancelar e botão de confirmar; *display* alfanumérico; um sistema de dinheiro com aceitador de moedas e cofre com fechadura eletrônica; um sistema de troco com 4 tubos para moedas e vasilha de troco; um sistema de estoque com compartimento para a retirada de produtos; uma porta traseira de manutenção trancada com chave e com um sensor de abertura; e 6 atuadores(motores de passo) para os helicoides que derrubam os produtos comprados no compartimento de retirada.

A máquina terá dois ciclos de operação: o de compra e o de manutenção. No ciclo de manutenção será feito o esvaziamento do cofre, a reposição do troco e dos produtos e a edição dos preços. Ao ser aberta a porta a porta de manutenção com a chave, o ciclo de operação mudará do de compra para o de manutenção e a máquina não poderá realizar operações de compra até que a porta seja novamente fechada e então a máquina retorne ao ciclo de compra. A chave de abertura da porta traseira será de posse de um técnico da empresa de manutenção da máquina. O mesmo também terá a senha do cofre onde ficam depositadas as moedas provenientes das compras efetuadas.

2 Ciclos de Operação

2.1 Ciclo Compra

2.1.1 Selecionar produto

O consumidor insere no teclado o código do produto que deseja e o mesmo será exibido no *display*. Caso o consumidor aperte o botão de cancelar a operação será cancelada.

2.1.2 Cancelar operação

Quando a operação é cancelada a máquina retorna ao seu estado inicial do ciclo de compra.

2.1.3 Devolver dinheiro

Caso a operação seja cancelada pelo usuário ou por falta de troco ou produto e tenham sido inseridas uma ou mais moedas, as mesmas serão devolvidas.

2.1.4 Inserir moedas

O usuário deverá inserir moedas até que a soma das mesmas iguale ou ultrapasse o valor do produto escolhido e selecionar o produto para prosseguir. Caso aperte o botão de cancelar ou não tenha inserido moedas o suficiente, a operação será cancelada.

2.1.5 Entregar produto

Após confirmado o pagamento, um motor de passo conectado ao arame helicoidal referente ao produto selecionado será acionado, empurrando o produto para a frente e depositando o mesmo no compartimento de retirada de produtos.

2.1.6 Entregar troco

Caso o pagamento necessite de troco (troco $\neq 0$), a máquina irá depositar as moedas necessárias na vasilha de troco.

2.2 Ciclo de Manutenção

2.2.1 Esvaziar cofre

Para coletar o dinheiro no cofre, o técnico deverá pressionar o botão 3 no teclado, seguido da senha numérica de 4 dígitos no teclado(senha padrão: 1234). Se a senha estiver correta, a porta do cofre, a qual é interna à porta traseira da máquina, será destravada. Caso esteja errada, a operação será cancelada.

2.2.2 Cadastrar produtos

Para cadastrar produtos (editar quantidade e preço), o técnico deverá pressionar o botão 2 no teclado. Então, o técnico deverá inserir a posição que deseja editar, seguida da nova quantidade e do novo valor(em centavos).

2.2.3 Repor troco

Para repor o troco, o técnico deverá pressionar o botão 1 no teclado. A máquina resetará o contador de cada tipo de moeda para seu máximo, 40 moedas de cada.

2.2.4 Entrar Manutenção

Ao abrir a porta traseira da *Vending Machine* com a chave de manutenção, o sensor da mesma é ativado e a máquina entra no ciclo de manutenção.

2.2.5 Sair Manutenção

Ao fechar a porta traseira da *Vending Machine* com a chave de manutenção, o sensor da mesma é desativado e a máquina retorna ao ciclo de compras.

3 Diagramas UML

Os Diagramas a seguir foram feitos no *software* Draw.io e representam de forma mais clara o funcionamento da *Vending Machine* que foi descrito nas sessões 1 e 2.

3.1 Casos de Uso

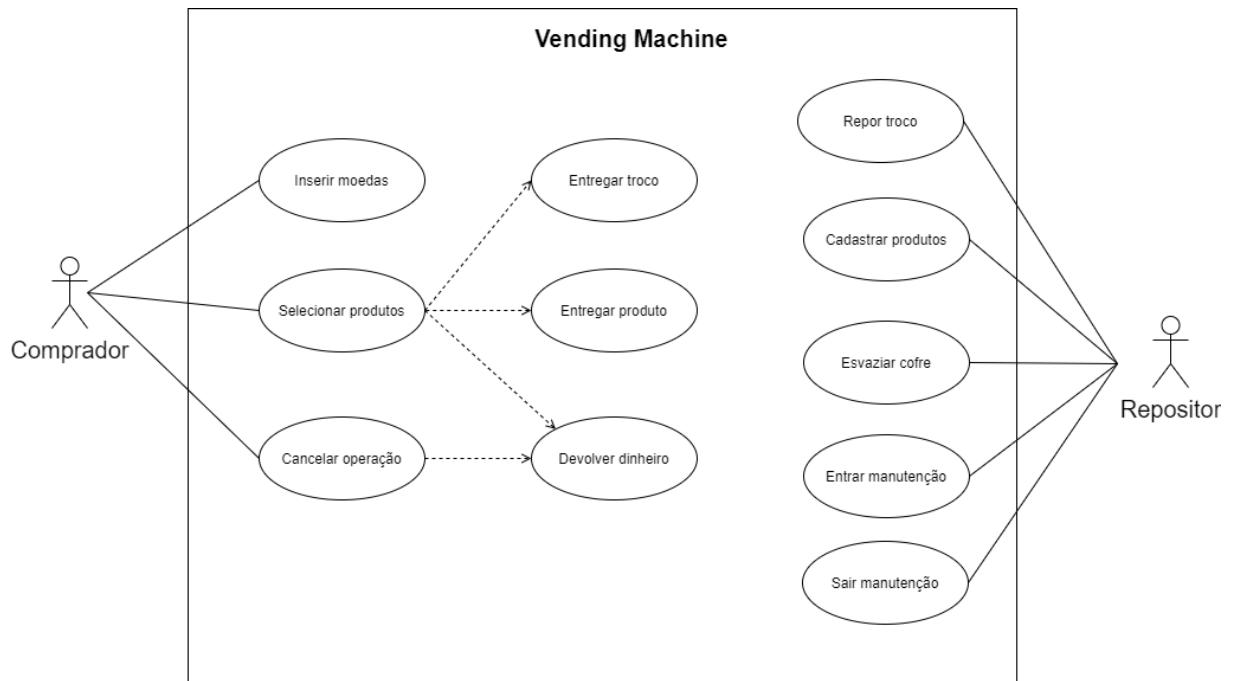


Figura 1: Diagrama de Casos de Uso da *Vending Machine*

3.2 Diagrama de Estados

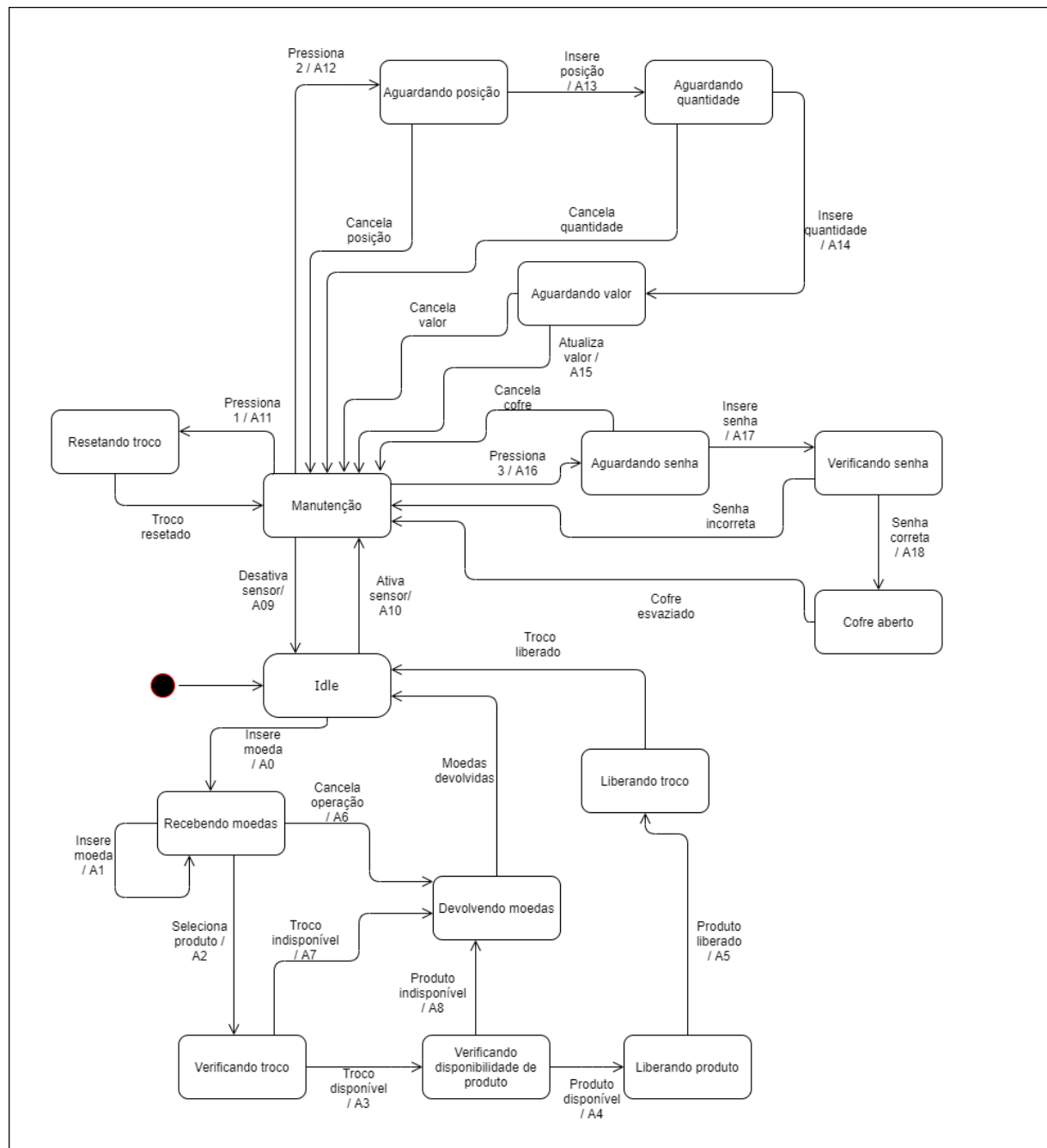


Figura 2: Diagrama de Estados da *Vending Machine*

A tabela abaixo lista as ações disparadas pelos eventos.

A00	soma valor, exibir saldo
A01	soma valor, exibir saldo
A02	seleciona produto, calcula troco, verifica troco
A03	verifica produto
A04	libera produto
A05	guarda dinheiro no cofre, libera troco
A06	exibe mensagem e devolve moedas
A07	exibe mensagem e devolve moedas
A08	exibe mensagem e devolve moedas
A09	desativa sensor
A010	ativa sensor
A011	reseta troco
A012	exibe mensagem (pede posição)
A013	seleciona posição, exibe mensagem (pede quantidade)
A014	atualiza quantidade, exibe mensagem (pede valor)
A015	atualiza valor
A016	exibe mensagem (pede senha)
A017	verifica senha
A018	abre cofre

Tabela 1: Lista de ações

3.3 Diagrama de Componentes

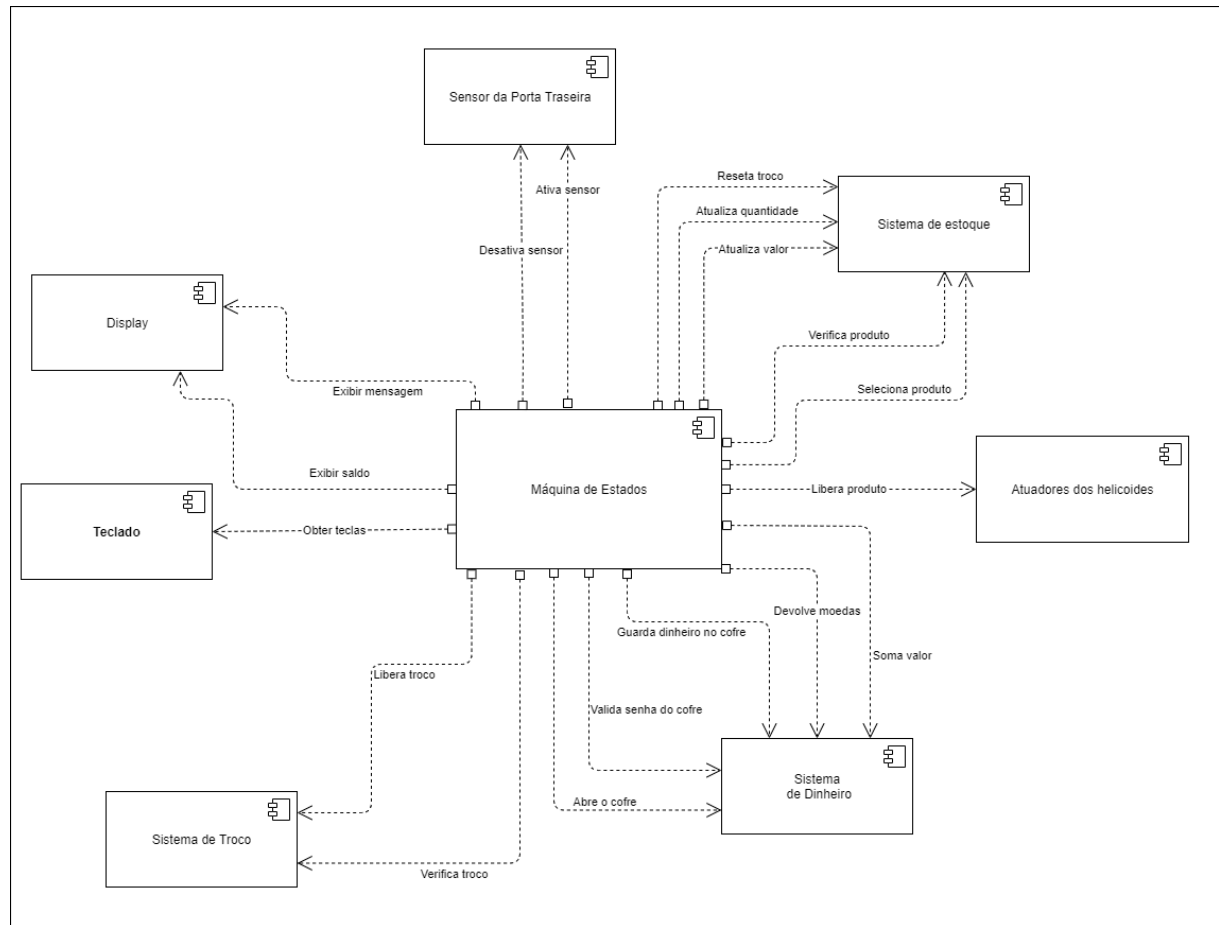


Figura 3: Diagrama de Componentes da *Vending Machine*

OBS: o *software* Draw.io não permite que seja inserido um quadrado na cabeça da seta como foi ensinado em sala de aula, apenas na extremidade da seta sem cabeça.

3.4 Diagramas de Sequência

Segue abaixo os diagramas de sequência para cada caso de uso ilustrado no diagrama da sessão 3.1. Para facilitar a compreensão casos de uso secundários foram representados no mesmo diagrama que seus respectivos casos de uso primários.

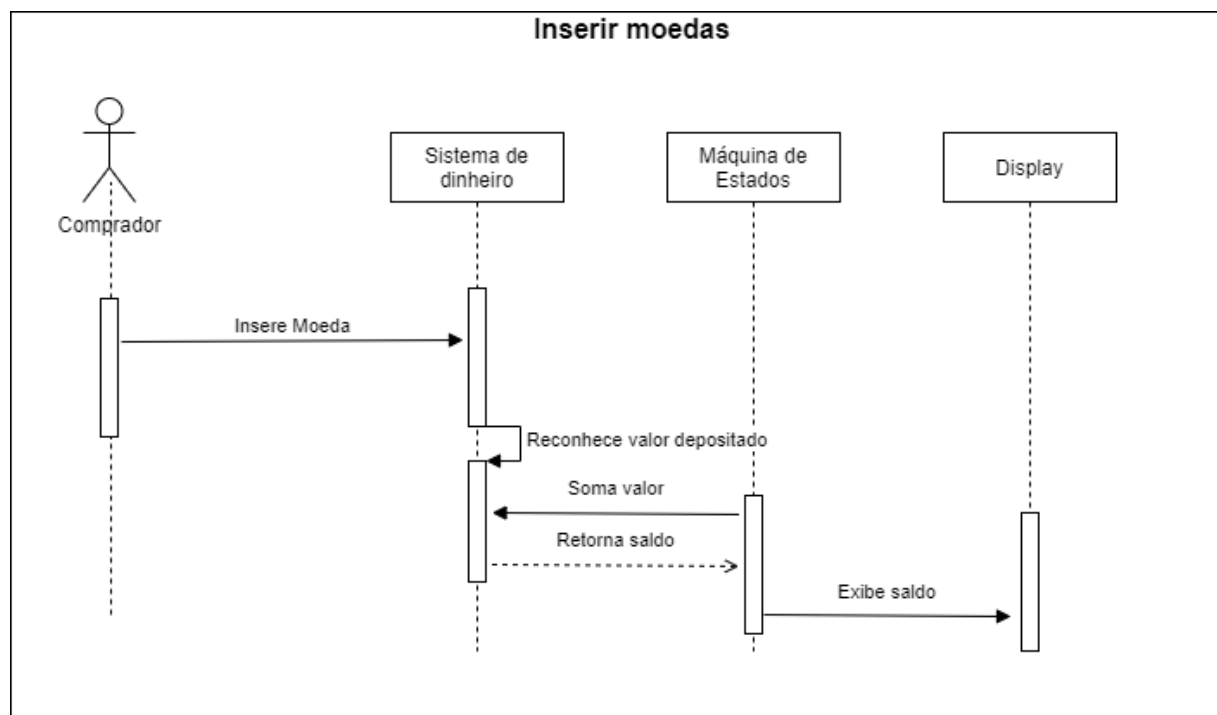


Figura 4: Inserir Moedas

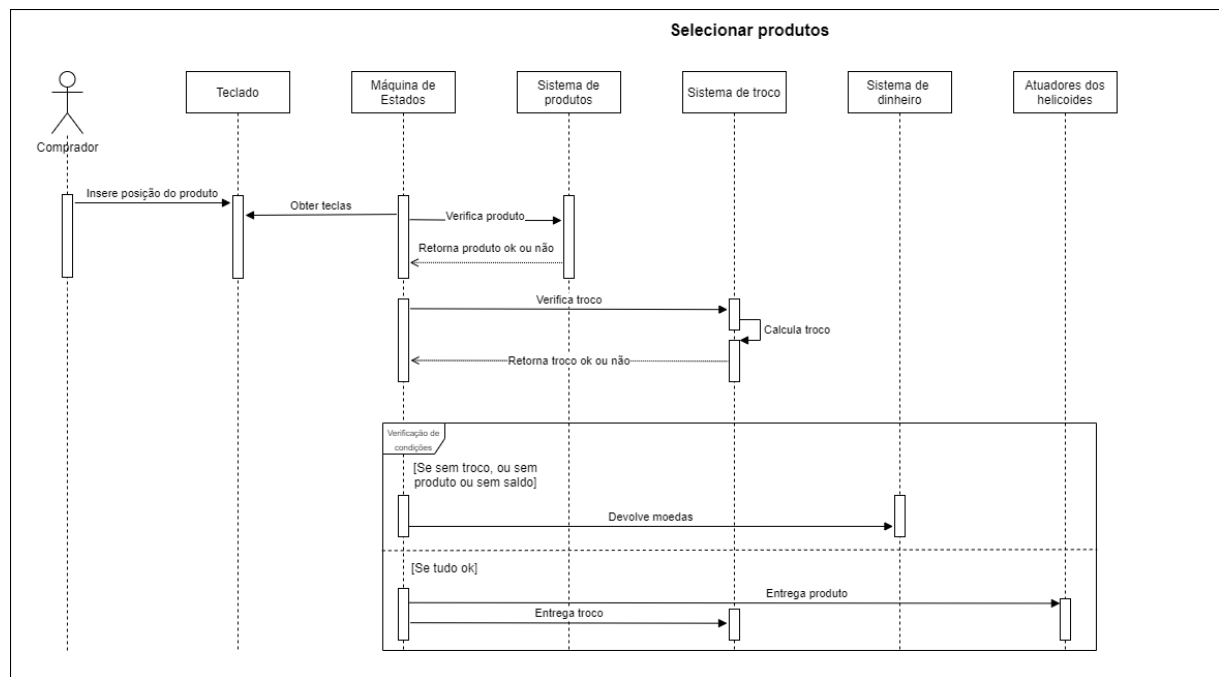


Figura 5: Selecionar Produtos

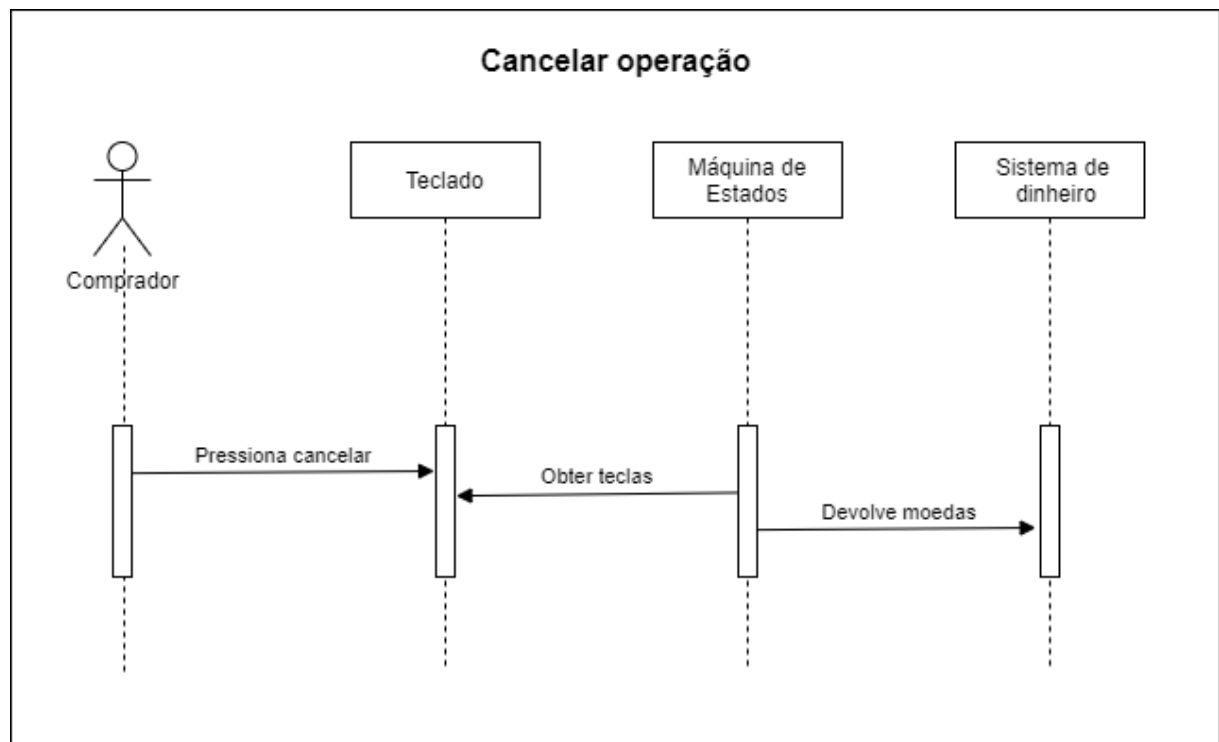


Figura 6: Cancelar Operação

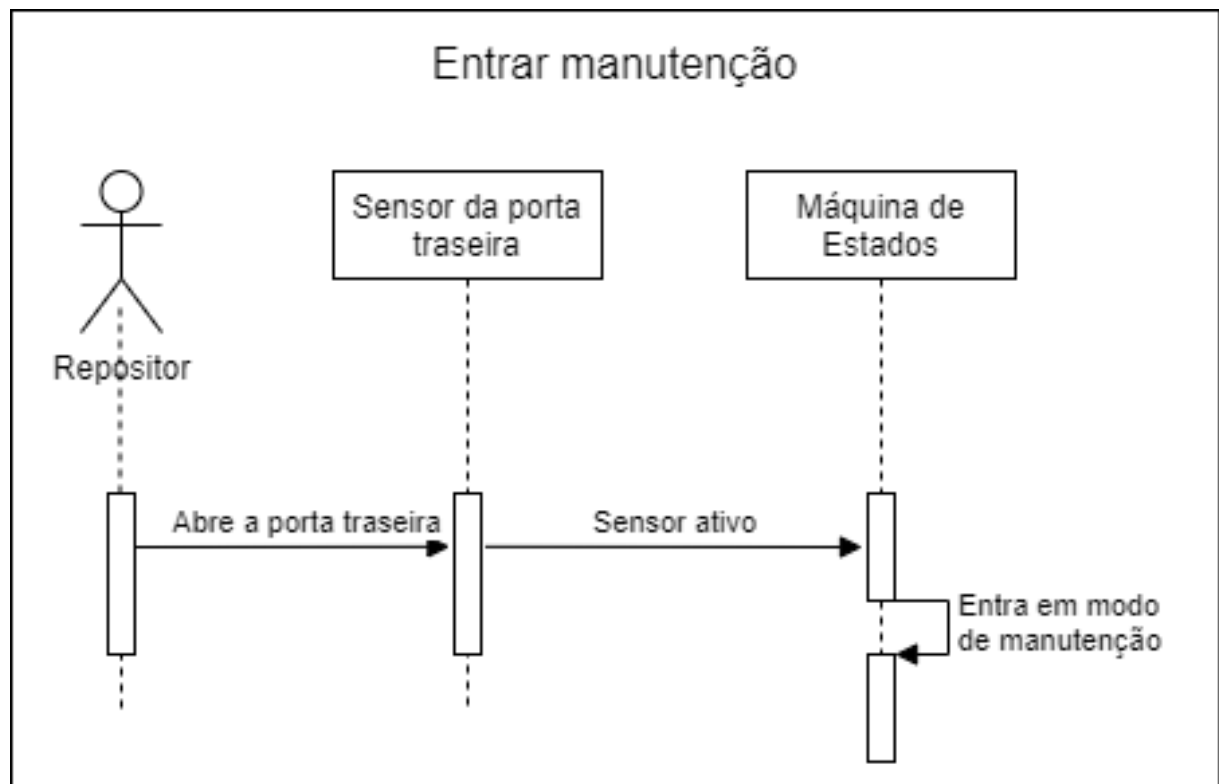


Figura 7: Entrar Manutenção

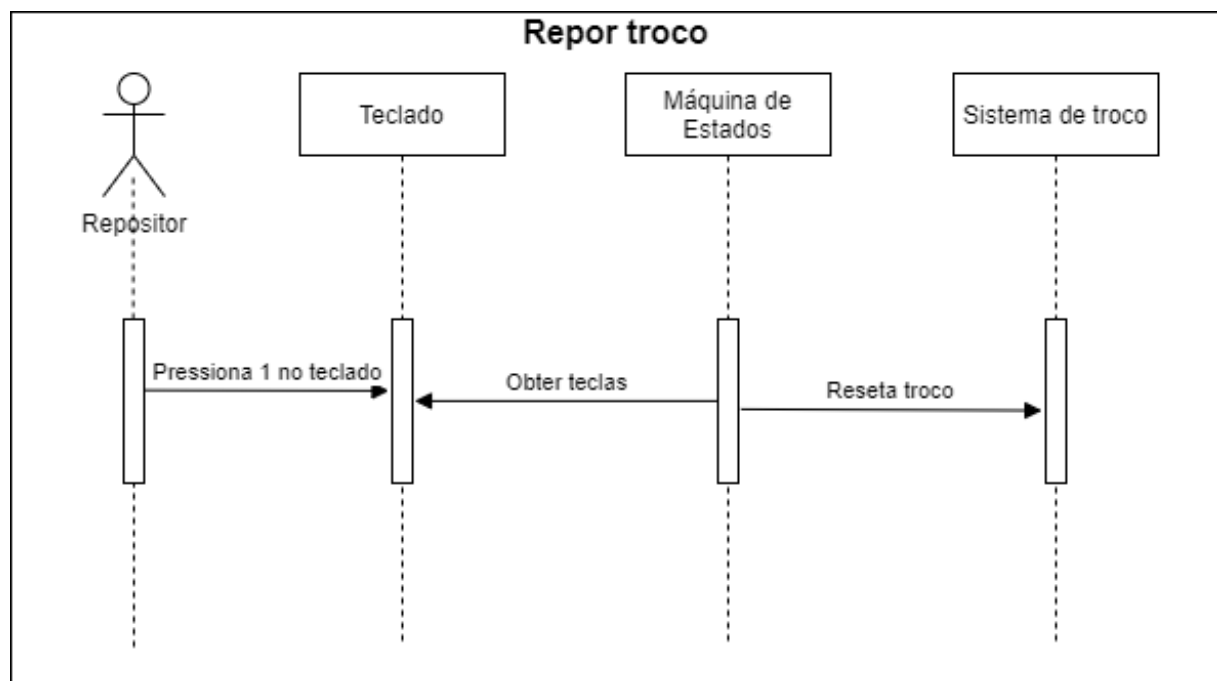


Figura 8: Repor Troco

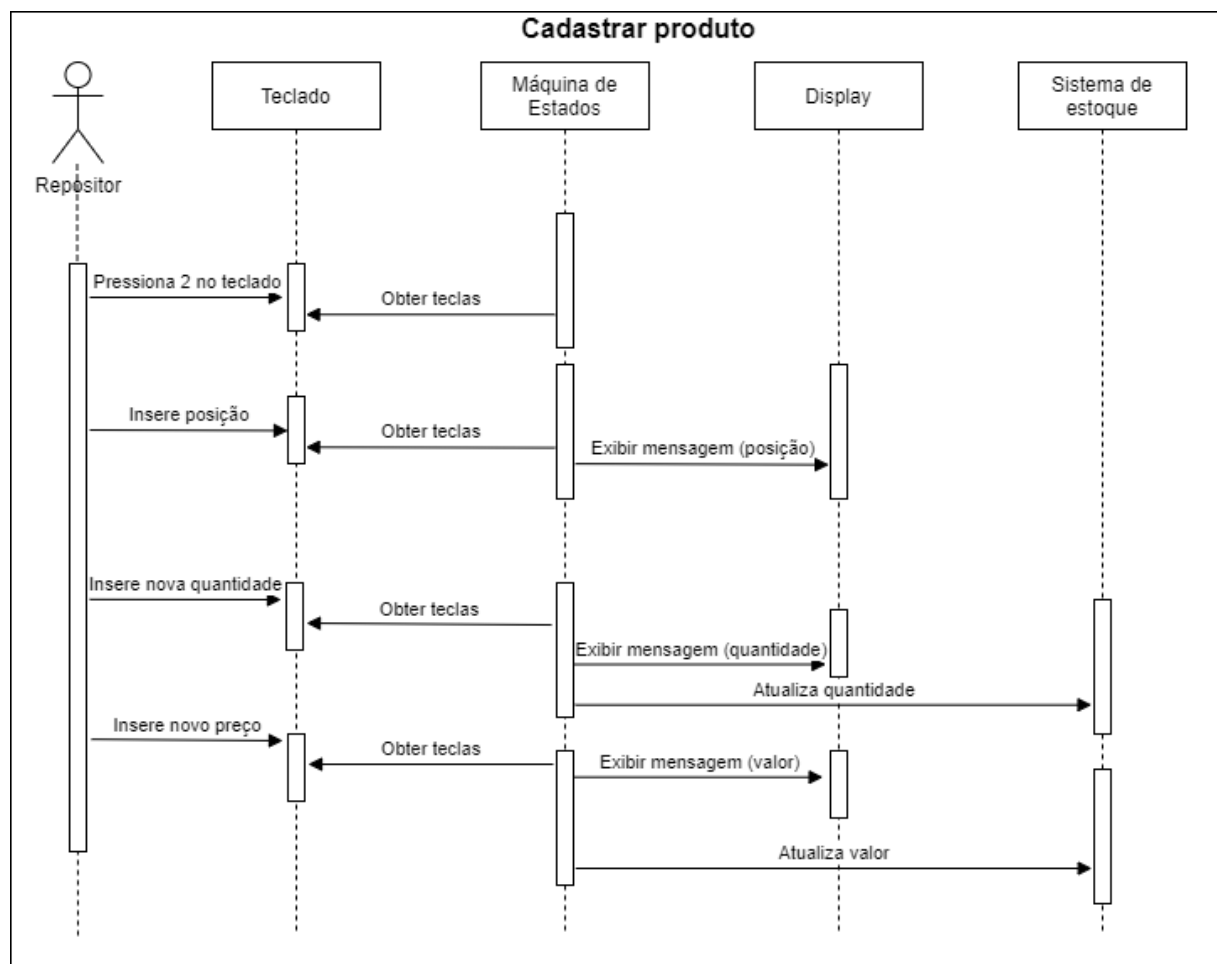


Figura 9: Cadastrar Produtos

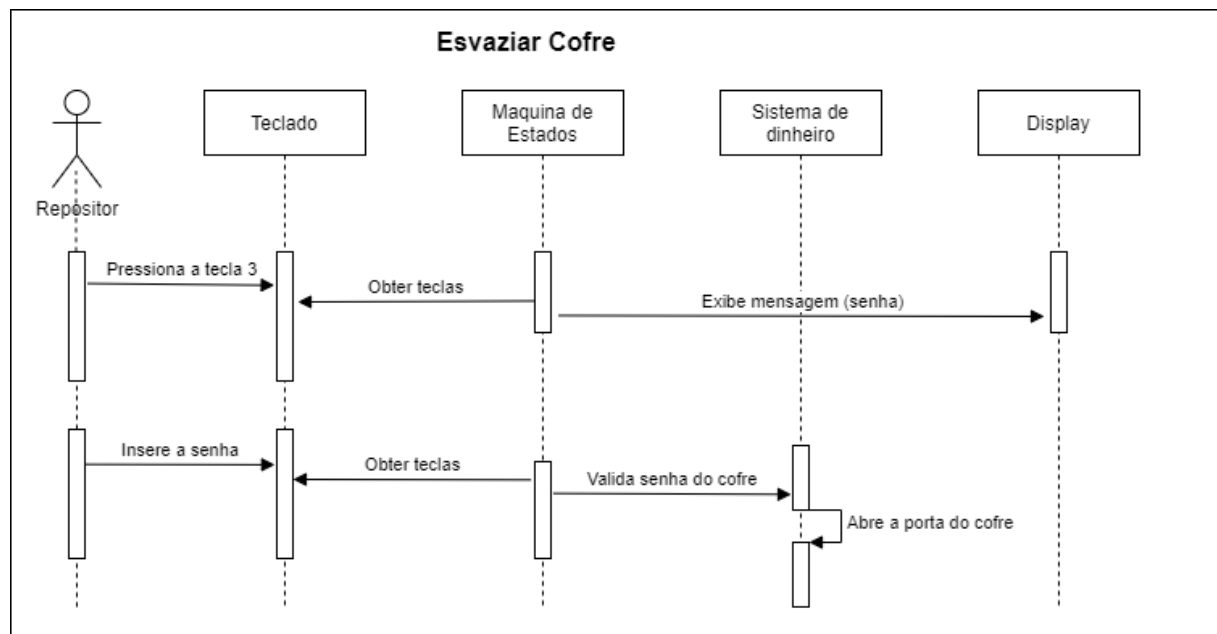


Figura 10: Esvaziar Cofre

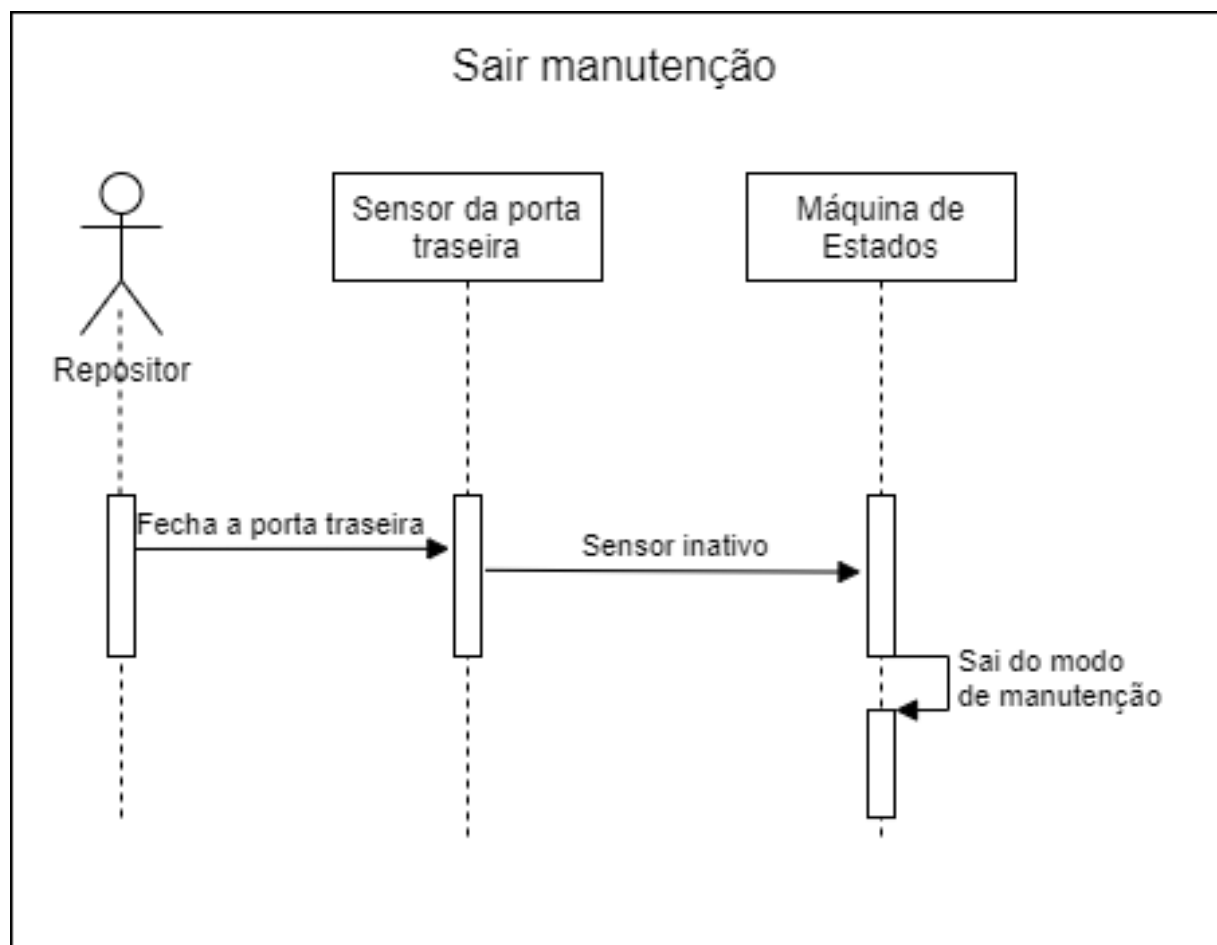


Figura 11: Sair Manutenção

4 Prints

Nesta sessão, serão exibidos alguns prints com as principais funcionalidades da *vending machine* sendo executada.

```
Vending Machine Iniciada
+++++
+      Vending Machine v1.0 - 11181038      +
+      Seja bem vindo!                      +
+      Codigos, produtos, quantidades e precos +
+ 01 - Pipoca Doce      - 5 unidades - R$ 3,00 +
+ 02 - Salgadinho       - 5 unidades - R$ 2,40 +
+ 03 - Chocolate        - 5 unidades - R$ 2,00 +
+ 04 - Amendoim Sal     - 5 unidades - R$ 1,30 +
+ 05 - Amendoim Doce    - 5 unidades - R$ 2,10 +
+ 06 - Balas            - 0 unidades - R$ 1,00 +
+++++
+ m xx - insere moeda de valor xx          +
+ s xx - seleciona o produto de codigo xx   +
+ c - cancela operacao                      +
+ M - modo manutencao                      +
+++++
obter teclas:m1
Saldo: R$ 1,00
Estado: 1 Evento: 0 Acao:0
obter teclas:m1
Saldo: R$ 2,00
Estado: 1 Evento: 0 Acao:1
obter teclas:m25
Saldo: R$ 2,25
Estado: 1 Evento: 0 Acao:1
obter teclas:s03
Voce selecionou Chocolate
Troco: R$ 0,25
Estado: 2 Evento: 1 Acao:2
Estado: 3 Evento: 2 Acao:3
Voce recebeu: Chocolate
Estado: 4 Evento: 3 Acao:4
Voce recebeu de troco 0 moedas de 1 real, 0 moedas de 50 centavos, 1 moedas de 25 centavos e 0 moedas de 10 centavos.
Estado: 5 Evento: 4 Acao:5
Estado: 0 Evento: 5 Acao:-1
obter teclas:
```

Figura 12: Compra realizada com sucesso.

```

+++++
+      Vending Machine v1.0 - 11181038      +
+      Seja bem vindo!                      +
+      Codigos, produtos, quantidades e precos  +
+      01 - Pipoca Doce      - 5 unidades - R$ 3,00 +
+      02 - Salgadinho       - 5 unidades - R$ 2,40 +
+      03 - Chocolate        - 5 unidades - R$ 2,00 +
+      04 - Amendoim Sal     - 5 unidades - R$ 1,30 +
+      05 - Amendoim Doce    - 5 unidades - R$ 2,10 +
+      06 - Balas            - 0 unidades - R$ 1,00 +
+++++
+ m xx - insere moeda de valor xx          +
+ s xx - seleciona o produto de codigo xx  +
+ c - cancela operacao                     +
+ M - modo manutencao                     +
+++++
obter teclas:m1
Saldo: R$ 1,00
Estado: 1 Evento: 0 Acao:0
obter teclas:s06
Voce selecionou Balas
Troco: R$ 0,00
Estado: 2 Evento: 1 Acao:2
Estado: 3 Evento: 2 Acao:3
Produto indisponível
Voce recebeu de volta R$1,00
Estado: 6 Evento: 8 Acao:8
Estado: 0 Evento: 9 Acao:-1
obter teclas:

```

Figura 13: Sem produto.

```

Vending Machine Iniciada

+++++
+      Vending Machine v1.0 - 11181038      +
+      Seja bem vindo!                      +
+      Codigos, produtos, quantidades e precos  +
+  01 - Pipoca Doce      - 5 unidades - R$ 3,00 +
+  02 - Salgadinho       - 5 unidades - R$ 2,40 +
+  03 - Chocolate        - 5 unidades - R$ 2,00 +
+  04 - Amendoim Sal     - 5 unidades - R$ 1,30 +
+  05 - Amendoim Doce    - 5 unidades - R$ 2,10 +
+  06 - Balas            - 0 unidades - R$ 1,00 +
+++++
+  m xx - insere moeda de valor xx          +
+  s xx - seleciona o produto de codigo xx   +
+  c - cancela operacao                      +
+  M - modo manutencao                      +
+++++
obter teclas:m1
Saldo: R$ 1,00
Estado: 1 Evento: 0 Acao:0
obter teclas:m1
Saldo: R$ 2,00
Estado: 1 Evento: 0 Acao:1
obter teclas:m1
Saldo: R$ 3,00
Estado: 1 Evento: 0 Acao:1
obter teclas:m50
Saldo: R$ 3,50
Estado: 1 Evento: 0 Acao:1
obter teclas:s05
Voce selecionou Amendoim Doce
Troco: R$ 1,40
Estado: 2 Evento: 1 Acao:2
Sem troco
Voce recebeu de volta R$3,50
Estado: 6 Evento: 7 Acao:7
Estado: 0 Evento: 9 Acao:-1
obter teclas:

```

Figura 14: Sem troco.

```

Vending Machine Iniciada

+++++
+      Vending Machine v1.0 - 11181038      +
+      Seja bem vindo!                      +
+      Codigos, produtos, quantidades e precos      +
+  01 - Pipoca Doce      - 5 unidades - R$ 3,00      +
+  02 - Salgadinho      - 5 unidades - R$ 2,40      +
+  03 - Chocolate      - 5 unidades - R$ 2,00      +
+  04 - Amendoim Sal    - 5 unidades - R$ 1,30      +
+  05 - Amendoim Doce   - 5 unidades - R$ 2,10      +
+  06 - Balas           - 0 unidades - R$ 1,00      +
+++++
+  m xx - insere moeda de valor xx      +
+  s xx - seleciona o produto de codigo xx      +
+  c - cancela operacao      +
+  M - modo manutencao      +
+++++
obter teclas:m1
Saldo: R$ 1,00
Estado: 1 Evento: 0 Acao:0
obter teclas:m1
Saldo: R$ 2,00
Estado: 1 Evento: 0 Acao:1
obter teclas:m1
Saldo: R$ 3,00
Estado: 1 Evento: 0 Acao:1
obter teclas:c
Operacao cancelada
Voce recebeu de volta R$3,00
Estado: 6 Evento: 6 Acao:6
Estado: 0 Evento: 9 Acao:-1
obter teclas:

```

Figura 15: Compra cancelada.

```

Vending Machine Iniciada

+++++
+      Vending Machine v1.0 - 11181038      +
+      Seja bem vindo!                      +
+      Codigos, produtos, quantidades e precos  +
+  01 - Pipoca Doce      - 5 unidades - R$ 3,00 +
+  02 - Salgadinho      - 5 unidades - R$ 2,40 +
+  03 - Chocolate       - 5 unidades - R$ 2,00 +
+  04 - Amendoim Sal    - 5 unidades - R$ 1,30 +
+  05 - Amendoim Doce   - 5 unidades - R$ 2,10 +
+  06 - Balas           - 0 unidades - R$ 1,00 +
+++++
+ m xx - insere moeda de valor xx          +
+ s xx - seleciona o produto de codigo xx  +
+ c - cancela operacao                     +
+ M - modo manutencao                      +
+++++
obter teclas:M
Sensor ativado, porta de manutencao aberta
+++++
+      Modo Manutencao                      +
+      Codigos, produtos, quantidades e precos  +
+  01 - Pipoca Doce      - 5 unidades - R$ 3,00 +
+  02 - Salgadinho      - 5 unidades - R$ 2,40 +
+  03 - Chocolate       - 5 unidades - R$ 2,00 +
+  04 - Amendoim Sal    - 5 unidades - R$ 1,30 +
+  05 - Amendoim Doce   - 5 unidades - R$ 2,10 +
+  06 - Balas           - 0 unidades - R$ 1,00 +
+++++
+ Saldo no cofre: R$78,10.                  +
+ Moedas para troco:                          +
+   3 de 1 real, 3 de 50 centavos,           +
+   3 de 25 centavos e 3 de 10 centavos.    +
+++++
+  1 - recarregar troco                      +
+  2 - editar quantidade/preço                +
+  3 - solicitar abertura do cofre            +
+                                             +
+++++

```

Figura 16: Modo de manutenção.


```

obter teclas:M
Sensor ativado, porta de manutencao aberta
+++++
+                               Modo Manutencao                               +
+   Codigos, produtos, quantidades e precos   +
+   01 - Pipoca Doce       - 5 unidades - R$ 3,00 +
+   02 - Salgadinho        - 5 unidades - R$ 2,40 +
+   03 - Chocolate         - 5 unidades - R$ 2,00 +
+   04 - Amendoim Sal      - 5 unidades - R$ 1,30 +
+   05 - Amendoim Doce     - 5 unidades - R$ 2,10 +
+   06 - Balas             - 0 unidades - R$ 1,00 +
+++++
+ Saldo no cofre: R$78,10. +
+ Moedas para troco: +
+   3 de 1 real, 3 de 50 centavos, +
+   3 de 25 centavos e 3 de 10 centavos. +
+++++
+ 1 - recarregar troco +
+ 2 - editar quantidade/preço +
+ 3 - solicitar abertura do cofre +
+ +
+++++
Estado: 7 Evento: 10 Acao:10
obter teclas:1
Troco resetado

+++++
+                               Modo Manutencao                               +
+   Codigos, produtos, quantidades e precos   +
+   01 - Pipoca Doce       - 5 unidades - R$ 3,00 +
+   02 - Salgadinho        - 5 unidades - R$ 2,40 +
+   03 - Chocolate         - 5 unidades - R$ 2,00 +
+   04 - Amendoim Sal      - 5 unidades - R$ 1,30 +
+   05 - Amendoim Doce     - 5 unidades - R$ 2,10 +
+   06 - Balas             - 0 unidades - R$ 1,00 +
+++++
+ Saldo no cofre: R$78,10. +
+ Moedas para troco: +
+   40 de 1 real, 40 de 50 centavos, +
+   40 de 25 centavos e 40 de 10 centavos. +
+++++
+ 1 - recarregar troco +
+ 2 - editar quantidade/preço +
+ 3 - solicitar abertura do cofre +
+ +
+++++
Estado: 8 Evento: 11 Acao:11
Estado: 7 Evento: 12 Acao:-1
obter teclas:

```

Figura 17: Repor troco.

```

+++++
+          Modo Manutencao          +
+  Codigos, produtos, quantidades e precos  +
+  01 - Pipoca Doce      - 5 unidades - R$ 3,00 +
+  02 - Salgadinho       - 5 unidades - R$ 2,40 +
+  03 - Chocolate        - 5 unidades - R$ 2,00 +
+  04 - Amendoim Sal     - 5 unidades - R$ 1,30 +
+  05 - Amendoim Doce    - 5 unidades - R$ 2,10 +
+  06 - Balas            - 0 unidades - R$ 1,00 +
+++++
+ Saldo no cofre: R$78,10.          +
+ Moedas para troco:                +
+   40 de 1 real, 40 de 50 centavos,  +
+   40 de 25 centavos e 40 de 10 centavos.  +
+++++
+ 1 - recarregar troco              +
+ 2 - editar quantidade/preço        +
+ 3 - solicitar abertura do cofre    +
+                                     +
+++++
Estado: 7 Evento: 10 Acao:10
obter teclas:2
Insira o codigo do produto que deseja editar:
Estado: 9 Evento: 13 Acao:12
obter teclas:06
Voce selecionou 06
Insira a nova quantidade:
Estado: 10 Evento: 14 Acao:13
obter teclas:9
Agora tem 9 unidades de Balas
Insira o novo valor(em centavos):
Estado: 11 Evento: 15 Acao:14
obter teclas:210
Agora R$ 2,10 é o novo valor de Balas

+++++
+          Modo Manutencao          +
+  Codigos, produtos, quantidades e precos  +
+  01 - Pipoca Doce      - 5 unidades - R$ 3,00 +
+  02 - Salgadinho       - 5 unidades - R$ 2,40 +
+  03 - Chocolate        - 5 unidades - R$ 2,00 +
+  04 - Amendoim Sal     - 5 unidades - R$ 1,30 +
+  05 - Amendoim Doce    - 5 unidades - R$ 2,10 +
+  06 - Balas            - 9 unidades - R$ 2,10 +
+++++
+ Saldo no cofre: R$78,10.          +
+ Moedas para troco:                +
+   40 de 1 real, 40 de 50 centavos,  +
+   40 de 25 centavos e 40 de 10 centavos.  +
+++++
+ 1 - recarregar troco              +
+ 2 - editar quantidade/preço        +
+ 3 - solicitar abertura do cofre    +
+                                     +
+++++
Estado: 7 Evento: 16 Acao:15
obter teclas:

```

Figura 18: Cadastrar produto.

```

+++++++
+                               +
+      Modo Manutencao      +
+  Codigos, produtos, quantidades e precos  +
+  01 - Pipoca Doce      - 5 unidades - R$ 3,00 +
+  02 - Salgadinho      - 5 unidades - R$ 2,40 +
+  03 - Chocolate      - 5 unidades - R$ 2,00 +
+  04 - Amendoim Sal    - 5 unidades - R$ 1,30 +
+  05 - Amendoim Doce   - 5 unidades - R$ 2,10 +
+  06 - Balas          - 9 unidades - R$ 2,10 +
+                               +
+ Saldo no cofre: R$78,10.      +
+ Moedas para troco:          +
+   40 de 1 real, 40 de 50 centavos,      +
+   40 de 25 centavos e 40 de 10 centavos.  +
+                               +
+ 1 - recarregar troco          +
+ 2 - editar quantidade/preço      +
+ 3 - solicitar abertura do cofre      +
+                               +
+ Estado: 7 Evento: 16 Acao:15
+ obter teclas:3
+ Insira a senha do cofre:
+ Estado: 12 Evento: 17 Acao:16
+ obter teclas:1234
+ senha correta
+ Estado: 13 Evento: 18 Acao:17
+ Porta do cofre aberta.
+ Voce retirou R$78,10
+
+++++++
+                               +
+      Modo Manutencao      +
+  Codigos, produtos, quantidades e precos  +
+  01 - Pipoca Doce      - 5 unidades - R$ 3,00 +
+  02 - Salgadinho      - 5 unidades - R$ 2,40 +
+  03 - Chocolate      - 5 unidades - R$ 2,00 +
+  04 - Amendoim Sal    - 5 unidades - R$ 1,30 +
+  05 - Amendoim Doce   - 5 unidades - R$ 2,10 +
+  06 - Balas          - 9 unidades - R$ 2,10 +
+                               +
+ Saldo no cofre: R$0,00.      +
+ Moedas para troco:          +
+   40 de 1 real, 40 de 50 centavos,      +
+   40 de 25 centavos e 40 de 10 centavos.  +
+                               +
+ 1 - recarregar troco          +
+ 2 - editar quantidade/preço      +
+ 3 - solicitar abertura do cofre      +
+                               +
+ Estado: 14 Evento: 19 Acao:18
+ Estado: 7 Evento: 21 Acao:-1
+ obter teclas:

```

Figura 19: Esvaziar cofre.


```

+++++
+          Modo Manutencao          +
+  Codigos, produtos, quantidades e precos  +
+  01 - Pipoca Doce      - 5 unidades - R$ 3,00 +
+  02 - Salgadinho      - 5 unidades - R$ 2,40 +
+  03 - Chocolate       - 5 unidades - R$ 2,00 +
+  04 - Amendoim Sal    - 5 unidades - R$ 1,30 +
+  05 - Amendoim Doce   - 5 unidades - R$ 2,10 +
+  06 - Balas           - 9 unidades - R$ 2,10 +
+++++
+ Saldo no cofre: R$0,00.          +
+ Moedas para troco:              +
+   40 de 1 real, 40 de 50 centavos,      +
+   40 de 25 centavos e 40 de 10 centavos.  +
+++++
+ 1 - recarregar troco              +
+ 2 - editar quantidade/preço        +
+ 3 - solicitar abertura do cofre    +
+                                  +
+++++
Estado: 14 Evento: 19 Acao:18
Estado: 7 Evento: 21 Acao:-1
obter teclas:3
Insira a senha do cofre:
Estado: 12 Evento: 17 Acao:16
obter teclas:1222
senha incorreta
Estado: 13 Evento: 18 Acao:17
Estado: 7 Evento: 20 Acao:-1
obter teclas:

```

Figura 20: Senha do cofre incorreta.

5 Conclusão

O uso de ferramentas UML ajuda o programador a entender o produto que o cliente deseja e corrigir muitos erros antes mesmo de começar a escrever seu código, reduzindo significativamente o tempo gasto na implementação.

Uma das maiores dificuldades que eu tive foi, durante a programação, ter que corrigir várias erros que encontrei nos diagramas UML tornando o processo iterativo e demorado. Porém imagino que um programador com prática em UML não teria esse problema e saberia reconhecer os erros nos seus diagramas antes de começar a programar, tirando grande proveito dessa poderosa ferramenta.

6 Código C

A *Vending Machine* foi programada em Linguagem C com um projeto estruturado em vários arquivos. Além de um arquivo `main.c` e um arquivo `definições_sistema.h`, foi criado para cada componente apresentado na figura 3(exceto para a máquina de estados, que é representada pelos arquivos `main.c` e `definições_sistema.h`) um arquivo `.c` contendo a definição das funções descritas na tabela 1 e um arquivo `.h` contendo as declarações dessas funções.

```
definicoes_sistema.h
1  #ifndef DEFINICOES_SISTEMA_H_INCLUDED
2  #define DEFINICOES_SISTEMA_H_INCLUDED
3
4  #define true 1
5  #define false 0
6
7  #define disponivel 1
8  #define indisponivel 1
9
10 #define correta 1
11 #define incorreta 1
12
13 #define NUM_ESTADOS 15
14 #define NUM_EVENTOS 27
15
16 // ESTADOS
17 #define IDLE 0
```

```

18 #define RECEBENDO_MOEDAS 1
19 #define VERIFICANDO_TROCO 2
20 #define VERIFICANDO_PRODUTO 3
21 #define LIBERANDO_PRODUTO 4
22 #define LIBERANDO_TROCO 5
23 #define DEVOLVENDO_MOEDAS 6
24 #define MANUTENCAO 7
25 #define RESETANDO_TROCO 8
26 #define AGUARDANDO_POSICAO 9
27 #define AGUARDANDO_QUANTIDADE 10
28 #define AGUARDANDO_VALOR 11
29 #define AGUARDANDO_SENHA 12
30 #define VERIFICANDO_SENHA 13
31 #define COFRE_ABERTO 14
32
33 // EVENTOS
34 #define NENHUM_EVENTO 1
35 #define INSERE_MOEDA 0
36 #define SELECIONA_PRODUTO 1
37 #define TROCO_DISPONIVEL 2
38 #define PRODUTO_DISPONIVEL 3
39 #define PRODUTO_LIBERADO 4
40 #define TROCO_LIBERADO 5
41 #define CANCELA_OPERACAO 6
42 #define TROCO_INDISPONIVEL 7
43 #define PRODUTO_INDISPONIVEL 8
44 #define MOEDAS_DEVOLVIDAS 9
45 #define ATIVA_SENSOR 10
46 #define PRESSIONA_1 11
47 #define TROCO_RESETADO 12
48 #define PRESSIONA_2 13
49 #define INSERE_POSICAO 14
50 #define INSERE_QUANTIDADE 15
51 #define ATUALIZA_VALOR 16
52 #define PRESSIONA_3 17
53 #define INSERE_SENHA 18
54 #define SENHA_CORRETA 19
55 #define SENHA_INCORRETA 20

```

```

56 #define COFRE_ESVAZIADO      21
57 #define CANCELA_POSICAO      22
58 #define CANCELA_QUANTIDADE    23
59 #define CANCELA_VALOR         24
60 #define CANCELA_COFRE         25
61 #define DESATIVA_SENSOR      26
62
63 // ACOES
64 #define NENHUMA_ACAO  1
65 #define A00  0
66 #define A01  1
67 #define A02  2
68 #define A03  3
69 #define A04  4
70 #define A05  5
71 #define A06  6
72 #define A07  7
73 #define A08  8
74 #define A09  9
75 #define A10 10
76 #define A11 11
77 #define A12 12
78 #define A13 13
79 #define A14 14
80 #define A15 15
81 #define A16 16
82 #define A17 17
83 #define A18 18
84
85
86 #endif // DEFINICOES_SISTEMA_H_INCLUDED

```

main.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5 /*

```

```

6      VENDING MACHINE
7  */
8
9  #include "definicoes_sistema.h"
10 #include "display.h"
11 #include "teclado.h"
12 #include "sistema_de_dinheiro.h"
13 #include "sistema_de_estoque.h"
14 #include "sistema_de_troco.h"
15 #include "sensor_da_porta.h"
16 #include "atuadores_dos_helicoides.h"
17
18
19
20 /*
21      *****
22
23  Estaticos
24      *****
25      */
26  int  codigoEvento;
27  int  codigoAcao;
28  int  estado;
29  int  sensores;
30  int  acao_matrizTransicaoEstados[NUM_ESTADOS][NUM_EVENTOS];
31  int  proximo_estado_matrizTransicaoEstados[NUM_ESTADOS][
32      NUM_EVENTOS];
33
34  /*
35      *****
36
37  executarAcao
38  Executa uma acao
39  Parametros de entrada:
40      (int) codigo da acao a ser executada
41  Retorno: (int) codigo do evento interno ou NENHUM_EVENTO
42      *****
43      */

```

```

37 char *teclas;
38 int executarAcao(int codigoAcao)
39 {
40     int retval;
41
42     retval = NENHUM_EVENTO;
43     if (codigoAcao == NENHUMA_ACAO)
44         return retval;
45
46     switch(codigoAcao)
47     {
48     case A00:
49         sdd_somaValor(teclas);
50         dpl_exibirSaldo();
51         break;
52     case A01:
53         sdd_somaValor(teclas);
54         dpl_exibirSaldo();
55         break;
56     case A02:
57         sde_selecionaProduto(teclas);
58         sdt_calculaTroco();
59         sdt_verificaTroco();
60         break;
61     case A03:
62         sde_verificaProduto();
63         break;
64     case A04:
65         adh_liberaProduto();
66         break;
67     case A05:
68         sdd_guardaDinheiro();
69         sdt_liberaTroco();
70         break;
71     case A06:
72         sdd_devolveMoedas();
73         break;
74     case A07:

```

```

75         sdd_devolveMoedas();
76         break;
77     case A08:
78         sdd_devolveMoedas();
79         break;
80     case A09:
81         sdp_desativaSensor();
82         break;
83     case A10:
84         sdp_ativaSensor();
85         break;
86     case A11:
87         sdt_resetaTroco();
88         break;
89     case A12:
90         dpl_exibirMsg("Insira o código do produto que deseja
91                       editar:\n");
92         //sde_pedePosicao();
93         break;
94     case A13:
95         sde_selecionaPosicao(teclas);
96         dpl_exibirMsg("Insira a nova quantidade:\n");
97         //sde_pedeQuantidade();
98         break;
99     case A14:
100        sde_atualizaQuantidade(teclas);
101        dpl_exibirMsg("Insira o novo valor(em centavos):\n");
102        //sde_pedeValor();
103        break;
104    case A15:
105        sde_atualizaValor(teclas);
106        break;
107    case A16:
108        dpl_exibirMsg("Insira a senha do cofre:\n");
109        //sdd_pedeSenha();
110        break;
111    case A17:
112        sdd_verificaSenha(teclas);

```

```

112         break;
113     case A18:
114         sdd_abreCofre();
115         break;
116     } // switch
117
118     return retval;
119 } // executarAcao
120
121 /*
122     *****
123
124     iniciaMaquina de Estados
125     Carrega a maquina de estados
126     Parametros de entrada: nenhum
127     Retorno: nenhum
128     *****
129     */
130 void iniciaMaquinaEstados()
131 {
132     int i;
133     int j;
134
135     for (i=0; i < NUM_ESTADOS; i++) {
136         for (j=0; j < NUM_EVENTOS; j++) {
137             acao_matrizTransicaoEstados[i][j] = NENHUMA_ACAO;
138             proximo_estado_matrizTransicaoEstados[i][j] = i;
139         }
140     }
141     proximo_estado_matrizTransicaoEstados[IDLE][INSERE_MOEDA] =
142         RECEBENDO_MOEDAS;
143     acao_matrizTransicaoEstados[IDLE][INSERE_MOEDA] = A00;
144
145     proximo_estado_matrizTransicaoEstados[RECEBENDO_MOEDAS][
146         INSERE_MOEDA] = RECEBENDO_MOEDAS;
147     acao_matrizTransicaoEstados[RECEBENDO_MOEDAS][INSERE_MOEDA]
148         = A01;
149
150 }

```



```

144 proximo_estado_matrizTransicaoEstados[RECEBENDO_MOEDAS][
    SELECIONA_PRODUTO] = VERIFICANDO_TROCO;
145 acao_matrizTransicaoEstados[RECEBENDO_MOEDAS][
    SELECIONA_PRODUTO] = A02;
146
147 proximo_estado_matrizTransicaoEstados[VERIFICANDO_TROCO][
    TROCO_DISPONIVEL] = VERIFICANDO_PRODUTO;
148 acao_matrizTransicaoEstados[VERIFICANDO_TROCO][
    TROCO_DISPONIVEL] = A03;
149
150 proximo_estado_matrizTransicaoEstados[VERIFICANDO_PRODUTO][
    PRODUTO_DISPONIVEL] = LIBERANDO_PRODUTO;
151 acao_matrizTransicaoEstados[VERIFICANDO_PRODUTO][
    PRODUTO_DISPONIVEL] = A04;
152
153 proximo_estado_matrizTransicaoEstados[LIBERANDO_PRODUTO][
    PRODUTO_LIBERADO] = LIBERANDO_TROCO;
154 acao_matrizTransicaoEstados[LIBERANDO_PRODUTO][
    PRODUTO_LIBERADO] = A05;
155
156 proximo_estado_matrizTransicaoEstados[LIBERANDO_TROCO][
    TROCO_LIBERADO] = IDLE;
157 acao_matrizTransicaoEstados[LIBERANDO_TROCO][TROCO_LIBERADO
    ] = NENHUMA_ACAO;
158
159 proximo_estado_matrizTransicaoEstados[RECEBENDO_MOEDAS][
    CANCELA_OPERACAO] = DEVOLVENDO_MOEDAS;
160 acao_matrizTransicaoEstados[RECEBENDO_MOEDAS][
    CANCELA_OPERACAO] = A06;
161
162 proximo_estado_matrizTransicaoEstados[VERIFICANDO_TROCO][
    TROCO_INDISPONIVEL] = DEVOLVENDO_MOEDAS;
163 acao_matrizTransicaoEstados[VERIFICANDO_TROCO][
    TROCO_INDISPONIVEL] = A07;
164
165 proximo_estado_matrizTransicaoEstados[VERIFICANDO_PRODUTO][
    PRODUTO_INDISPONIVEL] = DEVOLVENDO_MOEDAS;
166 acao_matrizTransicaoEstados[VERIFICANDO_PRODUTO][

```

```

167     PRODUTO_INDISPONIVEL] = A08;
168     proximo_estado_matrizTransicaoEstados[DEVOLVENDO_MOEDAS][
169         MOEDAS_DEVOLVIDAS] = IDLE;
170     acao_matrizTransicaoEstados[DEVOLVENDO_MOEDAS][
171         MOEDAS_DEVOLVIDAS] = NENHUMA_ACAO;
172     proximo_estado_matrizTransicaoEstados[IDLE][ATIVA_SENSOR] =
173         MANUTENCAO;
174     acao_matrizTransicaoEstados[IDLE][ATIVA_SENSOR] = A10;
175     proximo_estado_matrizTransicaoEstados[MANUTENCAO][
176         PRESSIONA_1] = RESETANDO_TROCO;
177     acao_matrizTransicaoEstados[MANUTENCAO][PRESSIONA_1] = A11
178         ;
179     proximo_estado_matrizTransicaoEstados[RESETANDO_TROCO][
180         TROCO_RESETADO] = MANUTENCAO;
181     acao_matrizTransicaoEstados[RESETANDO_TROCO][TROCO_RESETADO
182         ] = NENHUMA_ACAO;
183     proximo_estado_matrizTransicaoEstados[MANUTENCAO][
184         PRESSIONA_2] = AGUARDANDO_POSICAO;
185     acao_matrizTransicaoEstados[MANUTENCAO][PRESSIONA_2] = A12
186         ;
187     proximo_estado_matrizTransicaoEstados[AGUARDANDO_POSICAO][
188         INSERE_POSICAO] = AGUARDANDO_QUANTIDADE;
189     acao_matrizTransicaoEstados[AGUARDANDO_POSICAO][
190         INSERE_POSICAO] = A13;
191     proximo_estado_matrizTransicaoEstados[AGUARDANDO_QUANTIDADE
192         ][INSERE_QUANTIDADE] = AGUARDANDO_VALOR;
193     acao_matrizTransicaoEstados[AGUARDANDO_QUANTIDADE][
194         INSERE_QUANTIDADE] = A14;
195     proximo_estado_matrizTransicaoEstados[AGUARDANDO_VALOR][
196         ATUALIZA_VALOR] = MANUTENCAO;

```

```

190  acao_matrizTransicaoEstados [AGUARDANDO_VALOR] [
      ATUALIZA_VALOR] = A15;
191
192  proximo_estado_matrizTransicaoEstados [AGUARDANDO_POSICAO] [
      CANCELA_POSICAO] = MANUTENCAO;
193  acao_matrizTransicaoEstados [AGUARDANDO_POSICAO] [
      CANCELA_POSICAO] = NENHUMA_ACAO;
194
195  proximo_estado_matrizTransicaoEstados [AGUARDANDO_QUANTIDADE
      ] [CANCELA_QUANTIDADE] = MANUTENCAO;
196  acao_matrizTransicaoEstados [AGUARDANDO_QUANTIDADE] [
      CANCELA_QUANTIDADE] = NENHUMA_ACAO;
197
198  proximo_estado_matrizTransicaoEstados [AGUARDANDO_VALOR] [
      CANCELA_VALOR] = MANUTENCAO;
199  acao_matrizTransicaoEstados [AGUARDANDO_VALOR] [CANCELA_VALOR
      ] = NENHUMA_ACAO;
200
201  proximo_estado_matrizTransicaoEstados [MANUTENCAO] [
      DESATIVA_SENSOR] = IDLE;
202  acao_matrizTransicaoEstados [MANUTENCAO] [DESATIVA_SENSOR] =
      A09;
203
204  proximo_estado_matrizTransicaoEstados [MANUTENCAO] [
      PRESSIONA_3] = AGUARDANDO_SENHA;
205  acao_matrizTransicaoEstados [MANUTENCAO] [PRESSIONA_3] = A16
      ;
206
207  proximo_estado_matrizTransicaoEstados [AGUARDANDO_SENHA] [
      INSERE_SENHA] = VERIFICANDO_SENHA;
208  acao_matrizTransicaoEstados [AGUARDANDO_SENHA] [INSERE_SENHA]
      = A17;
209
210  proximo_estado_matrizTransicaoEstados [AGUARDANDO_SENHA] [
      CANCELA_COFRE] = MANUTENCAO;
211  acao_matrizTransicaoEstados [AGUARDANDO_SENHA] [CANCELA_COFRE
      ] = NENHUMA_ACAO;
212

```

```

213     proximo_estado_matrizTransicaoEstados[VERIFICANDO_SENHA][
        SENHA_CORRETA] = COFRE_ABERTO;
214     acao_matrizTransicaoEstados[VERIFICANDO_SENHA][
        SENHA_CORRETA] = A18;
215
216     proximo_estado_matrizTransicaoEstados[VERIFICANDO_SENHA][
        SENHA_INCORRETA] = MANUTENCAO;
217     acao_matrizTransicaoEstados[VERIFICANDO_SENHA][
        SENHA_INCORRETA] = NENHUMA_ACAO;
218
219     proximo_estado_matrizTransicaoEstados[COFRE_ABERTO][
        COFRE_ESVAZIADO] = MANUTENCAO;
220     acao_matrizTransicaoEstados[COFRE_ABERTO][COFRE_ESVAZIADO]
        = NENHUMA_ACAO;
221 } // initStateMachine
222
223 /*
    *****
224     iniciaSistema
225     Inicia o sistema ...
226     Parametros de entrada: nenhum
227     Retorno: nenhum
228     *****
    */
229 void iniciaSistema()
230 {
231     iniciaMaquinaEstados();
232     sde_populate();
233 } // initSystem
234
235
236 /*
    *****
237     obterEvento
238     Obtem um evento, que pode ser da IHM ou do alarme
239     Parametros de entrada: nenhum

```

```

240     Retorno: codigo do evento
241     ****
242     */
243
244
245     int decodificarInserirMoeda()
246     {
247         if (teclas[0] == 'm')
248         {
249             return true;
250         }
251         return false;
252     } // decodificarInserirMoeda
253     int decodificarSelecionarProduto()
254     {
255         if (teclas[0] == 's')
256         {
257             return true;
258         }
259         return false;
260     } // decodificarSelecionarProduto
261     int decodificarTrocoDisponivel()
262     {
263         if (flag_troco == disponivel)
264         {
265             return true;
266         }
267         else
268         {
269             return false;
270         }
271     }
272     int decodificarProdutoDisponivel()
273     {
274         if (flag_produto == disponivel)
275         {
276             return true;

```

```

277         }
278         return false;
279     }
280
281     int decodificarProdutoLiberado()
282     {
283         if (flag_produto_liberado==true)
284         {
285             return true;
286         }
287         return false;
288     }
289     int decodificarTrocoLiberado()
290     {
291         if (flag_troco_liberado==true)
292         {
293             return true;
294         }
295         return false;
296     }
297     int decodificarCancelaOperacao()
298     {
299         if (teclas[0] == 'c')
300         {
301             return true;
302         }
303         return false;
304     }
305     int decodificarTrocoIndisponivel()
306     {
307         if (flag_troco==indisponivel)
308         {
309             return true;
310         }
311         return false;
312     }
313     int decodificarProdutoIndisponivel()
314     {

```

```

315     if (flag_produto==indisponivel)
316     {
317         return true;
318     }
319     return false;
320 }
321 int decodificarMoedasDevolvidas()
322 {
323     if (flag_moedas_devolvidas == true)
324     {
325         return true;
326     }
327     return false;
328 }
329 int decodificarAtivaSensor()
330 {
331     if (teclas[0] == 'M')
332     {
333         return true;
334     }
335     return false;
336 }
337 int decodificarPressional()
338 {
339     if (teclas[0] == '1')
340     {
341         return true;
342     }
343     return false;
344 }
345 int decodificarTrocoResetado()
346 {
347     if (flag_troco_resetado == true)
348     {
349         return true;
350     }
351     return false;
352 }

```

```

353 int decodificarPressiona2()
354 {
355     if (teclas[0] == '2')
356     {
357         return true;
358     }
359     return false;
360 }
361 int decodificarInserePosicao()
362 {
363     if (teclas[0] != 'c')
364     {
365         return true;
366     }
367     return false;
368 }
369 int decodificarInsereQuantidade()
370 {
371     if (teclas[0] != 'c')
372     {
373         return true;
374     }
375     return false;
376 }
377 int decodificarAtualizaValor()
378 {
379     if (teclas[0] != 'c')
380     {
381         return true;
382     }
383     return false;
384 }
385 int decodificarPressiona3()
386 {
387     if (teclas[0] == '3')
388     {
389         return true;
390     }

```



```

391     return false;
392 }
393 int decodificarInsereSenha()
394 {
395     if (teclas[0] != 'c')
396     {
397         return true;
398     }
399     return false;
400 }
401 int decodificarSenhaCorreta()
402 {
403     if ((teclas[0] != 'c') && (flag_senha == correta))
404     {
405         return true;
406     }
407     return false;
408 }
409 int decodificarSenhaIncorreta()
410 {
411     if ((teclas[0] != 'c') && (flag_senha == incorreta))
412     {
413         return true;
414     }
415     return false;
416 }
417 int decodificarCofreEsvaziado()
418 {
419     if (flag_cofre_zerado == true)
420     {
421         return true;
422     }
423     return false;
424 }
425 int obterEvento()
426 {
427     int retval = NENHUM_EVENTO;
428

```

```

429     if (estado==IDLE || estado==RECEBENDO_MOEDAS || estado ==
        MANUTENCAO || estado == AGUARDANDO_POSICAO || estado
        == AGUARDANDO_QUANTIDADE || estado==AGUARDANDO_VALOR
        || estado == AGUARDANDO_SENHA){
430         teclas = tcl_obterTeclas();
431     }
432     else{
433         teclas = "z";
434     }
435
436
437     if (decodificarInserirMoeda())
438         return INSERE_MOEDA;
439     if (decodificarSelecionaProduto())
440         return SELECIONA_PRODUTO;
441     if ((decodificarTrocoDisponivel()) && (estado==
        VERIFICANDO_TROCO))
442         return TROCO_DISPONIVEL;
443     if (decodificarProdutoDisponivel() && (estado==
        VERIFICANDO_PRODUTO))
444         return PRODUTO_DISPONIVEL;
445     if ((decodificarProdutoLiberado()) && (estado==
        LIBERANDO_PRODUTO))
446         return PRODUTO_LIBERADO;
447     if ((decodificarTrocoLiberado()) && (estado==
        LIBERANDO_TROCO))
448         return TROCO_LIBERADO;
449     if ((decodificarCancelaOperacao()) && (estado==
        RECEBENDO_MOEDAS))
450         return CANCELA_OPERACAO;
451     if ((decodificarTrocoIndisponivel())&&(estado==
        VERIFICANDO_TROCO))
452         return TROCO_INDISPONIVEL;
453     if ((decodificarProdutoIndisponivel())&&(estado==
        VERIFICANDO_PRODUTO))
454         return PRODUTO_INDISPONIVEL;
455     if (decodificarMoedasDevolvidas())&&(estado==
        DEVOLVENDO_MOEDAS))

```

```

456     return MOEDAS_DEVOLVIDAS;
457 if ((decodificarAtivaSensor())&&(estado==IDLE))
458     return ATIVA_SENSOR;
459 if ((decodificarPressiona1())&&(estado==MANUTENCAO))
460     return PRESSIONA_1;
461 if ((decodificarTrocoResetado())&&(estado==RESETANDO_TROCO)
462     )
463     return TROCO_RESETADO;
464 if ((decodificarPressiona2())&&(estado==MANUTENCAO))
465     return PRESSIONA_2;
466 if ((decodificarInserePosicao())&&(estado==
467     AGUARDANDO_POSICAO))
468     return INSERE_POSICAO;
469 if ((decodificarInsereQuantidade())&&(estado==
470     AGUARDANDO_QUANTIDADE))
471     return INSERE_QUANTIDADE;
472 if ((decodificarAtualizaValor())&&(estado==AGUARDANDO_VALOR
473     ))
474     return ATUALIZA_VALOR;
475 if ((decodificarCancelaOperacao())&&(estado==
476     AGUARDANDO_POSICAO))
477     return CANCELA_POSICAO;
478 if ((decodificarCancelaOperacao())&&(estado==
479     AGUARDANDO_QUANTIDADE))
480     return CANCELA_QUANTIDADE;
481 if ((decodificarCancelaOperacao())&&(estado==
482     AGUARDANDO_VALOR))
483     return CANCELA_VALOR;
484 if ((decodificarAtivaSensor())&&(estado==MANUTENCAO))
485     return DESATIVA_SENSOR;
486 if ((decodificarPressiona3())&&(estado==MANUTENCAO))
487     return PRESSIONA_3;
488 if ((decodificarInsereSenha())&&(estado==AGUARDANDO_SENHA))
489     return INSERE_SENHA;
490 if ((decodificarCancelaOperacao())&&(estado==
491     AGUARDANDO_SENHA))
492     return CANCELA_COFRE;
493 if ((decodificarSenhaCorreta())&&(estado==VERIFICANDO_SENHA

```

```

    ))
486     return SENHA_CORRETA;
487 if ((decodificarSenhaIncorreta())&&(estado==
    VERIFICANDO_SENHA))
488     return SENHA_INCORRETA;
489 if ((decodificarCofreEsvaziado())&&(estado==COFRE_ABERTO))
490     return COFRE_ESVAZIADO;
491
492     return retval;
493
494 } // obterEvento
495
496 /*
    *****
497 obterAcao
498 Obtem uma acao da Matriz de transicao de estados
499 Parametros de entrada: estado (int)
500                        evento (int)
501 Retorno: codigo da acao
502 *****
    */
503 int obterAcao(int estado, int codigoEvento) {
504     return acao_matrizTransicaoEstados[estado][codigoEvento];
505 } // obterAcao
506
507
508 /*
    *****
509 obterProximoEstado
510 Obtem o proximo estado da Matriz de transicao de estados
511 Parametros de entrada: estado (int)
512                        evento (int)
513 Retorno: codigo do estado
514 *****
    */
515 int obterProximoEstado(int estado, int codigoEvento) {

```

```

516     return proximo_estado_matrizTransicaoEstados[estado][
        codigoEvento];
517 } // obterAcao
518
519
520 /*
        *****

521 Main
522 Loop principal de controle que executa a maquina de estados
523 Parametros de entrada: nenhum
524 Retorno: nenhum
525 *****
        */
526 int main() {
527
528     // int codigoEvento;
529     // int codigoAcao;
530     // int estado;
531     int eventoInterno;
532
533     estado = IDLE;
534     eventoInterno = NENHUM_EVENTO;
535
536     iniciaSistema();
537     printf ("Vending Machine Iniciada\n");
538     dpl_init();
539     while (true) {
540         if (eventoInterno == NENHUM_EVENTO) {
541             codigoEvento = obterEvento();
542         } else {
543             codigoEvento = eventoInterno;
544         }
545         if (codigoEvento != NENHUM_EVENTO)
546         {
547             codigoAcao = obterAcao(estado, codigoEvento);
548             estado = obterProximoEstado(estado, codigoEvento);
549             eventoInterno = executarAcao(codigoAcao);

```

```

550         printf("Estado: %d Evento: %d Acao:%d\n", estado ,
                codigoEvento , codigoAcao);
551     }
552 } // while true
553 } // main

```

atuadores_dos_helicoides.h

```

1  #ifndef ATUADORES_DOS_HELICOIDES_H_INCLUDED
2  #define ATUADORES_DOS_HELICOIDES_H_INCLUDED
3
4  /*****
5   adh_liberaProduto
6   Libera o produto comprado
7   entradas
8   nenhuma
9   saidas
10  mensagem com o nome do produto liberado
11  *****/
12  extern void adh_liberaProduto();
13
14  #endif // ATUADORES_DOS_HELICOIDES_H_INCLUDED

```

atuadores_dos_helicoides.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <locale.h>
5
6  #include "definicoes_sistema.h"
7  #include "sistema_de_estoque.h"
8  #include "atuadores_dos_helicoides.h"
9  #include "sistema_de_dinheiro.h"
10 #include "teclado.h"
11 #include "display.h"
12
13 /*****
14  adh_liberarProduto
15  Libera o produto comprado

```

```

16  entradas
17      nenhuma
18  saidas
19      mensagem com o nome do produto liberado
20  *****/
21 void adh_liberaProduto()
22 {
23     char msg[60];
24     sprintf(msg, "Voce recebeu: %s \n", produto[selecionado].
        nome);
25     dpl_exibirMsg(msg);
26     produto[selecionado].quantidade =1;
27     flag_produto_liberado=true;
28 }

```

display.h

```

1  #ifndef DISPLAY_H_INCLUDED
2  #define DISPLAY_H_INCLUDED
3
4  /******
5   dpl_exibirSaldo
6   Exibe saldo no display
7   entradas
8       int: saldo
9   saidas
10      print com o saldo
11  *****/
12 extern void dpl_exibirSaldo();
13
14  /******
15   dpl_exibirMsg
16   Exibe uma mensagem de outro componente
17   entradas
18       mensagem desejada
19   saidas
20      print com a mensagem desejada
21  *****/
22 extern void dpl_exibirMsg(char* msg);

```

```

23
24 /*****
25  dpl_init
26  Exibe os produtos disponiveis no display
27  entradas
28      nenhuma
29  saidas
30      print com produtos, preo e quantidade
31 *****/
32 extern void dpl_init();
33
34 /*****
35  dpl_initManutencao
36  Exibe os produtos disponiveis no display,
37  bem como o saldo no cofre e o troco
38  disponvel
39  entradas
40      nenhuma
41  saidas
42      print com produtos, preo, quantidade
43      saldo no cofre e troco disponvel
44 *****/
45 extern void dpl_initManutencao();
46
47 #endif // DISPLAY_H_INCLUDED

```

display.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <locale.h>
5
6 #include "definicoes_sistema.h"
7 #include "display.h"
8 #include "sistema_de_dinheiro.h"
9 #include "sistema_de_estoque.h"
10 #include "sistema_de_troco.h"
11

```



```

12
13 /*****
14 dpl_exibirSaldo
15 Exibe saldo no display
16 entradas
17     int: saldo
18 saidas
19     print com o saldo
20 *****/
21 void dpl_exibirSaldo()
22 {
23     printf("Saldo: R$ %d,%.2d \n", saldo/100,saldo%100);
24 }
25
26 /*****
27 dpl_exibirMsg
28 Exibe uma mensagem de outro componente
29 entradas
30     mensagem desejada
31 saidas
32     print com a mensagem desejada
33 *****/
34 void dpl_exibirMsg(char* msg)
35 {
36     printf(msg, setlocale(LC_ALL, ""));
37 }
38
39 /*****
40 dpl_init
41 Exibe os produtos disponiveis no display
42 entradas
43     nenhuma
44 saidas
45     print com produtos, preo e quantidade
46 *****/
47 void dpl_init()
48 {
49     ; int i;

```

```

50     printf("\n
        ++++++
        n");
51     printf("+      Vending Machine v1.0      11181038
        +\n");
52     printf("+                      Seja bem vindo!
        +\n");
53     printf("+      Codigos, produtos, quantidades e
        precos      +\n");
54     for(i = 0; i < 6; i++){
55         printf("+      0%d      %s      %d unidades      R$ %d
        ,%.2d      +\n",i+1, produto[i].nome, produto[i
        ].quantidade, produto[i].valor/100,produto[i
        ].valor%100);
56     }
57     printf("
        ++++++
        n");
58     printf("+  m xx      insere moeda de valor xx
        +\n");
59     printf("+  s xx      seleciona o produto de codigo
        xx      +\n");
60     printf("+  c      cancela operacao
        +\n");
61     printf("+  M      modo manutencao
        +\n");
62     printf("
        ++++++
        n");
63 }
64
65 /*****
66 dpl_initManutencao
67 Exibe os produtos disponiveis no display,
68 bem como o saldo no cofre e o troco
69 disponvel
70 entradas
71 nenhuma

```

```

72  saidas
73  print com produtos, preo, quantidade
74  saldo no cofre e troco disponvel
75  *****/
76  void dpl_initManutencao()
77  {
78  ;   int i;
79
      printf("\n
          ++++++
          n");
80  printf("+          Modo Manutencao
          +\n");
81  printf("+   Codigos, produtos, quantidades e
          precos   +\n");
82  for(i = 0; i < 6; i++){
83      printf("+   0%d   %s   %d unidades   R$ %d
          ,%.2d   +\n",i+1, produto[i].nome, produto[i
          ].quantidade, produto[i].valor/100,produto[i
          ].valor%100);
84  }
85  printf("
          ++++++
          n");
86  printf("+ Saldo no cofre: R$%d,%.2d.
          +\n",cofre/100,cofre
          %100);
87  printf("+ Moedas para troco:
          +\n");
88  printf("+   %d de 1 real, %d de 50 centavos,
          +\n",moedas_para_troco[0],
          moedas_para_troco[1]);
89  printf("+   %d de 25 centavos e %d de 10
          centavos.   +\n",moedas_para_troco[2],
          moedas_para_troco[3]);
90  printf("
          ++++++
          n");
91  printf("+   1   recarregar troco

```

```

92         printf("+ 2  editar quantidade/preo
           +\n");
93         printf("+ 3  solicitar abertura do cofre
           +\n");
94         printf("+
           +\n");
95         printf("
           ++++++
           n");
96     }

```

```

                                     sensor_da_porta.h
1  #ifndef SENSOR_DA_PORTA_H_INCLUDED
2  #define SENSOR_DA_PORTA_H_INCLUDED
3
4  /*****
5   sdp_ativaSensor
6   Entra em modo de manuten quando
7   o sensor de porta aberta  ativado
8   entradas
9   nenhuma
10  saidas
11  mensagem de sensor ativado
12  *****/
13  extern void sdp_ativaSensor();
14
15  /*****
16  sdp_desativaSensor
17  Sai do modo de manuten quando
18  o sensor de porta aberta  desativado
19  entradas
20  nenhuma
21  saidas
22  mensagem de sensor desativado
23  *****/
24  extern void sdp_desativaSensor();

```

```

25 #endif // SENSOR_DA_PORTA_H_INCLUDED

sensor_da_porta.c

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #include "definicoes_sistema.h"
6 #include "display.h"
7 #include "sistema_de_dinheiro.h"
8 #include "sistema_de_estoque.h"
9 #include "sistema_de_troco.h"
10 #include "sensor_da_porta.h"
11
12 /*****
13 sdp_ativaSensor
14 Entra em modo de manuten quando
15 o sensor de porta aberta ativado
16 entradas
17 nenhuma
18 saidas
19 mensagem de sensor ativado
20 *****/
21 void sdp_ativaSensor()
22 {
23     dpl_exibirMsg("Sensor ativado, porta de manuten aberta")
24     ;
25     dpl_initManutencao();
26 }
27 /*****
28 sdp_desativaSensor
29 Sai do modo de manuten quando
30 o sensor de porta aberta desativado
31 entradas
32 nenhuma
33 saidas
34 mensagem de sensor desativado

```

```

35  *****/
36  void sdp_desativaSensor()
37  {
38      dpl_exibirMsg("Sensor desativado , porta de manuten
          fechada");
39      dpl_init();
40  }

```

```

                                     sistema_de_dinheiro.h
1  #ifndef SISTEMA_DE_DINHEIRO_H_INCLUDED
2  #define SISTEMA_DE_DINHEIRO_H_INCLUDED
3
4  /**
5   Variveis Globais
6  *****/
7  extern int saldo;
8  extern int cofre;
9  extern int flag_moedas_devolvidas;
10 extern int flag_senha;
11 extern int flag_cofre_zerado;
12
13 /**
14  sdd_somaValor
15  Soma ao saldo o valor da moeda inserida
16  entradas
17  texto : texto a ser validado como moeda
18  saidas
19  saldo
20  *****/
21 extern void sdd_somaValor(char* texto);
22
23 /**
24  sdd_guardaDinheiro
25  Soma ao cofre o valor do saldo e zera o
26  saldo depois
27  entradas
28  nenhuma
29  saidas

```

```

30     saldo e cofre
31     *****/
32 extern void sdd_guardaDinheiro();
33
34     /*
35 sdd_devolverMoedas
36     devolve as moedas inseridas e zera o saldo
37     entradas
38     nenhuma
39     saidas
40     saldo e mensagem com as moedas devolvidas
41     e com o motivo da devolu
42     *****/
43 extern void sdd_devolveMoedas();
44
45     /*
46 sdd_verificaSenha
47     verifica a senha do cofre
48     entradas
49     texto: senha inserida pelo usuario
50     saidas
51     mensagem com senha correta ou incorreta
52     *****/
53 extern void sdd_verificaSenha(char* texto);
54
55     /*
56 sdd_abreCofre
57     abre a porta do cofre e zera o saldo no mesmo
58     entradas
59     nenhuma
60     saidas
61     mensagem com porta do cofre aberta e quantia
62     retirada
63     *****/
64 extern void sdd_abreCofre();
65
66 #endif // SISTEMA_DE_DINHEIRO_H_INCLUDED

```

sistema_de_dinheiro.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #include "definicoes_sistema.h"
5  #include "sistema_de_dinheiro.h"
6  #include "sistema_de_troco.h"
7  #include "sistema_de_estoque.h"
8  #include "display.h"
9
10 int saldo=0;
11 int cofre=7810;
12 int flag_troco;
13 int flag_moedas_devolvidas;
14 int flag_senha;
15 int flag_cofre_zerado;
16
17 /*****
18  sdd_somaValor
19  Soma ao saldo o valor da moeda inserida
20  entradas
21  texto : texto a ser validado como moeda
22  saidas
23  saldo
24  *****/
25 void sdd_somaValor(char* texto)
26 {
27     flag_troco=false;
28     flag_moedas_devolvidas=false;
29     if (texto[1] == '1' && texto[2] == '0')
30         saldo += 10;
31     if (texto[1] == '2' && texto[2] == '5')
32         saldo += 25;
33     if (texto[1] == '5' && texto[2] == '0')
34         saldo += 50;
35     if (texto[1] == '1' && texto[2] == '\0')
36         saldo += 100;
37 }

```



```

38
39 /*****
40 sdd_guardaDinheiro
41   Soma ao cofre o valor do saldo e zera o
42   saldo depois
43   entradas
44   nenhuma
45   saidas
46   saldo e cofre
47 *****/
48 void sdd_guardaDinheiro()
49 {
50     cofre += saldo;
51     saldo = 0;
52 }
53
54 /*****
55 sdd_devolverMoedas
56   devolve as moedas inseridas e zera o saldo
57   entradas
58   nenhuma
59   saidas
60   saldo e mensagem com as moedas devolvidas
61   e com o motivo da devolu
62 *****/
63 void sdd_devolveMoedas()
64 {
65     char msg[60];
66     if (troco < 0){
67         dpl_exibirMsg("Saldo insuficiente \n");
68     }
69     else{
70         if (produto[selecionado].quantidade==0){
71             dpl_exibirMsg("Produto indisponivel \n");
72         }
73         else{
74             if (flag_troco==indisponivel){
75                 dpl_exibirMsg("Sem troco \n");

```

```

76         }
77         else{
78             dpl_exibirMsg("Operacao cancelada \n");
79         }
80     }
81 }
82 sprintf(msg,"Voce recebeu de volta R$%d,%.2d \n", saldo
83         /100,saldo%100);
84 dpl_exibirMsg(msg);
85 saldo = 0;
86 flag_moedas_devolvidas=true;
87 }
88 /*****
89 sdd_verificaSenha
90     verifica a senha do cofre
91     entradas
92     texto: senha inserida pelo usuario
93     saidas
94     mensagem com senha correta ou incorreta
95 *****/
96 void sdd_verificaSenha(char* texto)
97 {
98     char msg[60];
99     if (texto[0]=='1'&&texto[1]=='2'&&texto[2]=='3'&&texto
100        [3]=='4'){
101         flag_senha=correta;
102         sprintf(msg,"senha correta\n");
103         dpl_exibirMsg(msg);
104     }
105     else{
106         flag_senha=incorreta;
107         sprintf(msg,"senha incorreta\n");
108         dpl_exibirMsg(msg);
109     }
110 }
111

```

```

112  /*****
113  sdd_abreCofre
114  abre a porta do cofre e zera o saldo no mesmo
115  entradas
116  nenhuma
117  saidas
118  mensagem com porta do cofre aberta e quantia
119  retirada
120  *****/
121  void sdd_abreCofre()
122  {
123      char msg[60];
124      sprintf(msg, "Porta do cofre aberta. \n");
125      dpl_exibirMsg(msg);
126      sprintf(msg, "Voce retirou R$%d,%.2d \n", cofre/100, cofre
127              %100);
128      dpl_exibirMsg(msg);
129      cofre=0;
130      flag_cofre_zerado=true;
131      dpl_initManutencao();
132  }

```

sistema_de_estoque.h

```

1  #ifndef SISTEMA_DE_ESTOQUE_H_INCLUDED
2  #define SISTEMA_DE_ESTOQUE_H_INCLUDED
3
4
5  /*****
6  Variaveis Globais
7  *****/
8  struct dados_de_produto
9  {
10     char nome[13];
11     int valor;
12     int quantidade;
13 };
14 extern struct dados_de_produto produto[6];
15 extern int selecionado;

```

```

16 extern int flag_produto;
17 extern int flag_produto_liberado;
18 extern int flag_quantidade_atualizada;
19 extern int flag_valor_atualizado;
20
21 /*****
22  sdd_populate
23  Cria o banco de dados inicial de produtos
24  entradas
25      nenhuma
26  saidas
27      nome, preco e quantidade de cada produto
28 *****/
29 extern void sde_populate();
30
31 /*****
32  sde_selecionarProduto
33  seleciona o produto a ser comprado
34  entradas
35      texto: teclas do teclado
36  saidas
37      mensagem com nome do produto selecionado
38      ou c d i g o   i n v l i d o
39 *****/
40 extern void sde_selecionaProduto(char* texto);
41
42 /*****
43  sde_verificarProduto
44  verifica se o produto a ser selecionado
45  est disponvel
46  entradas
47      nenhuma
48  saidas
49      nenhuma ou mensagem com produto indisponvel
50 *****/
51 extern void sde_verificaProduto();
52
53 /*****

```

```

54 sde_selecionaPosicao(char* texto)
55     seleciona a posi do produto a ser editado
56     entradas
57     texto: teclas do teclado
58     saidas
59     mensagem com produto selecionado ou
60     c digo i n v lido
61     *****/
62 extern void sde_selecionaPosicao(char* texto);
63
64 /******
65 sde_atualizaQuantidade
66     atualiza a quantidade do produto a ser editado
67     entradas
68     nenhuma
69     saidas
70     nova quantidade e mensagem com a mesma
71     *****/
72 extern void sde_atualizaQuantidade(char* texto);
73
74 /******
75 sde_atualizaValor
76     atualiza o valor do produto a ser editado
77     entradas
78     nenhuma
79     saidas
80     novo valor e mensagem com o mesmo
81     *****/
82 extern void sde_atualizaValor(char* texto);
83
84
85 #endif // SISTEMA_DE_ESTOQUE_H_INCLUDED

```

sistema_de_estoque.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <locale.h>

```

```

5
6 #include "definicoes_sistema.h"
7 #include "sistema_de_estoque.h"
8 #include "sistema_de_dinheiro.h"
9 #include "teclado.h"
10 #include "display.h"
11
12 int selecionado;
13 int flag_produto;
14 int flag_produto_liberado;
15 int flag_quantidade_atualizada;
16 int flag_valor_atualizado;
17 struct dados_de_produto produto[6];
18
19 /*****
20  sdd_populate
21  Cria o banco de dados inicial de produtos
22  entradas
23  nenhuma
24  saidas
25  nome, preo e quantidade de cada produto
26 *****/
27 void sde_populate()
28 {
29     strcpy(produto[0].nome, "Pipoca Doce ");
30     produto[0].valor=300;
31     produto[0].quantidade=5;
32     strcpy(produto[1].nome, "Salgadinho ");
33     produto[1].valor=240;
34     produto[1].quantidade=5;
35     strcpy(produto[2].nome, "Chocolate ");
36     produto[2].valor=200;
37     produto[2].quantidade=5;
38     strcpy(produto[3].nome, "Amendoim Sal ");
39     produto[3].valor=130;
40     produto[3].quantidade=5;
41     strcpy(produto[4].nome, "Amendoim Doce");
42     produto[4].valor=210;

```

```

43     produto[4].quantidade=5;
44     strcpy(produto[5].nome,"Balas");
45     produto[5].valor=100;
46     produto[5].quantidade=0;
47 }
48
49 /*****
50 sde_selecionarProduto
51     seleciona o produto a ser comprado
52     entradas
53     texto: teclas do teclado
54     saidas
55     mensagem com nome do produto selecionado
56     ou código inválido
57 *****/
58 void sde_selecionaProduto(char* texto)
59 {
60     char msg[60];
61     flag_produto=false;
62     flag_produto_liberado=false;
63     char selecionado_char[3] = {texto[1],texto[2]};
64     int codigo_selecionado = strtol(selecionado_char,NULL,10)
65     1;
66     if (codigo_selecionado > 1 && codigo_selecionado <6){
67         selecionado=codigo_selecionado;
68         sprintf(msg,"Voce selecionou %s \n",produto[
69             selecionado].nome);
70         dpl_exibirMsg(msg);
71     }
72     else{
73         sprintf(msg,"Codigo invalido.\n");
74         dpl_exibirMsg(msg);
75         tcl_obterTeclas();
76     }
77 }
78
79 /*****
80 sde_verificarProduto

```

```

79  verifica se o produto a ser selecionado
80  est disponvel
81  entradas
82      nenhuma
83  saidas
84      nenhuma
85  *****/
86  void sde_verificaProduto()
87  {
88      char msg[60];
89      if (produto[selecionado].quantidade>=1 && flag_produto==
          false)
90      {
91          flag_produto=disponivel;
92      }
93      else{
94          flag_produto=indisponivel;
95      }
96  }
97
98  /******
99  sde_selecionaPosicao(char* texto)
100  seleciona a posi do produto a ser editado
101  entradas
102      texto: teclas do teclado
103  saidas
104      mensagem com produto selecionado ou
105      c digo i n v lido
106  *****/
107  void sde_selecionaPosicao(char* texto)
108  {
109      char msg[60];
110      char selecionado_char[3] = {texto[1],texto[2]};
111      int codigo_selecionado = strtol(selecionado_char,NULL,10)
          1;
112      if (codigo_selecionado > 1 && codigo_selecionado <6){
113          selecionado=codigo_selecionado;
114          sprintf(msg,"Voce selecionou 0%d \n",selecionado+1);

```



```

115         dpl_exibirMsg(msg);
116     }
117     else{
118         sprintf(msg,"Codigo invalido.\n");
119         dpl_exibirMsg(msg);
120         tcl_obterTeclas();
121     }
122 }
123
124 /*****
125 sde_atualizaQuantidade
126     atualiza a quantidade do produto a ser editado
127     entradas
128     nenhuma
129     saidas
130     nova quantidade e mensagem com a mesma
131 *****/
132 void sde_atualizaQuantidade(char* texto)
133 {
134     char msg[60];
135     char quantidade_char[3] = {texto[0],texto[1]};
136     int quantidade = strtol(quantidade_char,NULL,10);
137     produto[selecionado].quantidade=quantidade;
138     flag_quantidade_atualizada=true;
139     sprintf(msg,"Agora tem %d unidades de %s \n",produto[
        selecionado].quantidade,produto[selecionado].nome);
140     dpl_exibirMsg(msg);
141 }
142
143 /*****
144 sde_atualizaValor
145     atualiza o valor do produto a ser editado
146     entradas
147     nenhuma
148     saidas
149     novo valor e mensagem com o mesmo
150 *****/
151 void sde_atualizaValor(char* texto)

```

```

152 {
153     char msg[60];
154     char valor_char[4] = {texto[0], texto[1], texto[2]};
155     int valor = strtol(valor_char, NULL, 10);
156     produto[selecionado].valor=valor;
157     flag_valor_atualizado=true;
158     sprintf(msg, "Agora R$ %d,%.2d    o novo valor de %s \n",
        produto[selecionado].valor/100, produto[selecionado].
        valor%100, produto[selecionado].nome, setlocale(LC_ALL, "
        "));
159     dpl_exibirMsg(msg);
160     dpl_initManutencao();
161 }

```

sistema_de_troco.h

```

1  #ifndef SISTEMA_DE_TROCO_H_INCLUDED
2  #define SISTEMA_DE_TROCO_H_INCLUDED
3
4  /*****
5   Variaveis Globais
6   *****/
7  extern int flag_troco;
8  extern int troco;
9  extern int flag_troco_liberado;
10 extern int flag_troco_resetado;
11 extern int moedas_para_troco[4];
12
13 /*****
14 sdt_calculaTroco
15     calcula o troco
16     entradas
17     nenhuma
18     saidas
19     valor do troco
20 *****/
21 extern void sdt_calculaTroco();
22
23 /*****

```

```

24 sdt_verificaTroco
25     verifica se h troco disponvel
26     entradas
27     nenhuma
28     saidas
29     quantidade de cada moeda para o troco
30     ou flag de troco indisponvel
31     *****/
32 extern void sdt_verificaTroco();
33
34 /******
35 sdt_liberaTroco
36     subtrai o troco da quantidade total de moedas
37     para troco na mquina e libera o troco
38     entradas
39     nenhuma
40     saidas
41     nova quantidade de moedas para troco
42     disponveis e moedas para troco liberadas
43     *****/
44 extern void sdt_liberaTroco();
45
46 /******
47 sdt_resetaTroco
48     reseta o troco disponvel para o mximo, 40
49     moedas de cada tipo
50     entradas
51     nenhuma
52     saidas
53     nova quantidade de moedas para troco
54     disponveis
55     *****/
56 extern void sdt_resetaTroco();
57
58 #endif // SISTEMA_DE_TROCO_H_INCLUDED

```

sistema_de_troco.c

```

1 #include <stdio.h>

```

```

2  #include <stdlib.h>
3  #include <string.h>
4
5  #include "definicoes_sistema.h"
6  #include "sistema_de_troco.h"
7  #include "sistema_de_dinheiro.h"
8  #include "sistema_de_estoque.h"
9  #include "display.h"
10
11  int moedas_para_troco[4]={3,3,3,3}; // 1 real, 50, 25 e 10
        centavos
12  int troco;
13  int flag_troco;
14  int flag_troco_liberado;
15  int flag_troco_resetado;
16  int c1=0;
17  int c2=0;
18  int c3=0;
19  int c4=0;
20
21  /*****
22  sdt_calculaTroco
23  calcula o troco
24  entradas
25  nenhuma
26  saidas
27  valor do troco
28  *****/
29  void sdt_calculaTroco()
30  {
31      char msg[60];
32      troco = saldo_produto[selecionado].valor;
33      if (troco>=0){
34          sprintf(msg,"Troco: R$ %d,%.2d \n", troco/100,troco
              %100);
35          dpl_exibirMsg(msg);
36      }
37  }

```

```

38
39
40 /*****
41 sdt_verificaTroco
42   verifica se h troco disponvel
43   entradas
44   nenhuma
45   saidas
46   quantidade de cada moeda para o troco
47   ou flag de troco indisponvel
48 *****/
49 void sdt_verificaTroco()
50 {
51     int qtd1=moedas_para_troco[0];
52     int qtd2=moedas_para_troco[1];
53     int qtd3=moedas_para_troco[2];
54     int qtd4=moedas_para_troco[3];
55     flag_troco_liberado = false;
56     while ( (troco>0) && ((troco    100 > 5) || (troco    100 ==
        0)) && (qtd1>0) )
57     {
58         troco  = 100;
59         qtd1  =1;
60         c1  +=1;
61     }
62     while ( (troco>0) && ((troco    50 > 5) || (troco    50 == 0))
        && (qtd2>0) )
63     {
64         troco  = 50;
65         qtd2  =1;
66         c2  +=1;
67     }
68     while ( (troco>0) && ((troco    25 > 5) || (troco    25 == 0))
        && (qtd3>0) )
69     {
70         troco  = 25;
71         qtd3  =1;
72         c3  +=1;

```

```

73     }
74     while ( ( troco>0) && (troco    10 >= 0) )
75     {
76         troco = 10;
77         qtd4  =1;
78         c4  +=1;
79     }
80     if ( ( troco==0) && (qtd4>=0) && (flag_troco == false) )
81     {
82         flag_troco = disponivel;
83     }
84     else
85     {
86         flag_troco = indisponivel;
87     }
88 }
89
90 /*****
91 sdt_liberaTroco
92 subtrai o troco da quantidade total de moedas
93 para troco na mquina e libera o troco
94 entradas
95     nenhuma
96 saidas
97     nova quantidade de moedas para troco
98     disponveis e moedas para troco liberadas
99 *****/
100 void sdt_liberaTroco()
101 {
102     char msg[200];
103     sprintf(msg,"Voce recebeu de troco %d moedas de 1 real , %
        d moedas de 50 centavos, %d moedas de 25 centavos e %d
        moedas de 10 centavos.\n", c1 ,c2 ,c3 ,c4 );
104     dpl_exibirMsg(msg);
105     moedas_para_troco[0] = c1;
106     moedas_para_troco[1] = c2;
107     moedas_para_troco[2] = c3;
108     moedas_para_troco[3] = c4;

```

```

109     flag_troco_liberado=true;
110 }
111
112 /*****
113 sdt_resetaTroco
114   reseta o troco disponvel para o mximo, 40
115   moedas de cada tipo
116   entradas
117   nenhuma
118   saidas
119   nova quantidade de moedas para troco
120   disponveis
121 *****/
122 void sdt_resetaTroco()
123 {
124     int i;
125     for(i = 0; i <= 3; i++)
126         moedas_para_troco[i]=40;
127     flag_troco_resetado=true;
128     dpl_exibirMsg("Troco resetado \n");
129     dpl_initManutencao();
130 }

```

teclado.h

```

1  #ifndef TECLADO_H_INCLUDED
2  #define TECLADO_H_INCLUDED
3
4  #define NENHUMA_TECLA 1
5
6  /*****
7   tcl_obterTeclas
8   Obtem tecla do teclado
9   entradas
10   nenhuma
11   saidas
12   teclas lidas do teclado ou NENHUMA_TECLA
13 *****/
14 extern char* tcl_obterTeclas();

```

```

15
16 #endif // TECLADO_H_INCLUDED

                                teclado.c

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #include "definicoes_sistema.h"
5  #include "teclado.h"
6
7  /*****
8   tcl_obterTeclas
9   Obtem tecla do teclado
10  entradas
11   nenhuma
12  saidas
13   teclas lidas do teclado ou NENHUMA_TECLA
14  *****/
15  char buf[10];
16  char* tcl_obterTeclas()
17  {
18      printf("obter teclas:");
19      scanf("%s", buf);
20      return buf;
21  }

```