UNIVERSIDADE da MADEIRA

2022/2023

FCEE – LEI

ARQUITETURA DE COMPUTADORES

Díonisio Barros | Dino Vasconcelos | Pedro Camacho | Sofia Inácio

TRABALHO PRÁTICO 2
MÁQUINA AUTOMÁTICA DE VENDAS

NUNO FERNANDES - 2082021 | JOÃO CASTRO - 2079521

ÍNDICE

INTRODUÇÃO

Este relatório descreve o segundo trabalho prático da disciplina de Arquitetura de Computadores, que envolveu o desenvolvimento de um programa na linguagem *Assembly* do processador PEPE. O mesmo inclui informações sobre os objetivos do trabalho, o processo de desenvolvimento, os resultados obtidos e as conclusões alcançadas pelos alunos.

A linguagem *Assembly* é específica para cada processador e permite que as instruções sejam codificadas em um único número, chamado de opcode.

O programa foi convertido em números binários pelo assembler, permitindo que o processador execute as instruções diretamente. Para testar o programa, foi utilizado um simulador em JAVA.

OBJETIVOS

O objetivo deste trabalho é criar um programa, em linguagem *Assembly*, que simula uma máquina de vendas. Elaborar e especificar fluxogramas é também um objetivo deste trabalho.

O stock da máquina apenas está disponível a alguém com acesso à palavra-passe.

O processo de venda é realizado através da escolha de um produto em primeiro lugar para depois ser possível inserir o dinheiro para pagar o produto escolhido recebendo o troco caso exista.

A interface com o utilizador foi realizada através de um display (periférico de saída), de dimensões 7x16 (7 linhas de 16 carateres – bytes). A interação do utilizador com o sistema é efetuado através de um periférico de entrada denominado *PER_EN*, um botão de ligar a máquina *ON_OFF* e um *Opcao* para escolher uma opcao mostrada nos menus.

DESENVOLVIMENTO

--PERIFÉRICOS DE ENTRADA--

ON_OFF

Este botão é responsável por ligar e desligar a máquina automática de vendas. Os utilizadores só podem utilizar a máquina quando o periférico estiver diferente de 0.

Caso esteja a “0”, a máquina do posto é desligada. Quando a máquina está desligada o display fica em branco até o utilizador a ligar.

Opcao

Este periférico serve para o utilizador inserir a opção para os diferentes menus.

Caso o utilizador escolha uma opção que não esteja disponível no display e confirme a escolha, ele será informado de que a opção escolhida é inválida.

PER_EN

Este periférico é utilizado para inserir a palavra-passe ou o dinheiro dependendo do que é pedido no decorrer do programa.

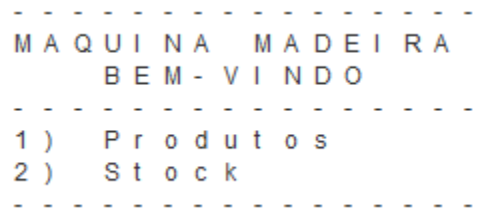
Caso a palavra passe ou o dinheiro estejam incorretos ele mostra um menu de erro.

--PROGRAMA--

Neste projeto alguns registos foram utilizados estaticamente para guardar determinados valore:

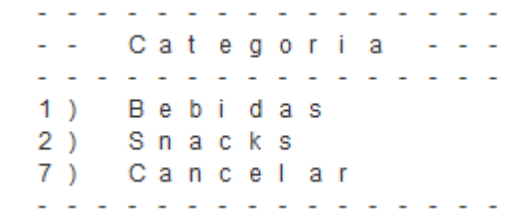
- R2 é utilizado para guardar o Menu ativo
- R5 é utilizado para guardar o Endereço do Stock
- R7 é utilizado para guardar o Endereço do Dinheiro Total Inserido
- R9 é utilizado para guardar o Endereço do último Dinheiro que foi inserido

Inicialmente a máquina encontra-se desligada. Após ser ligada apresenta o menu principal que está guardado em R2. Este fica à espera que o utilizador selecione uma das opções disponíveis.

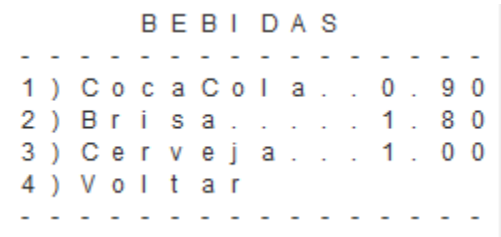


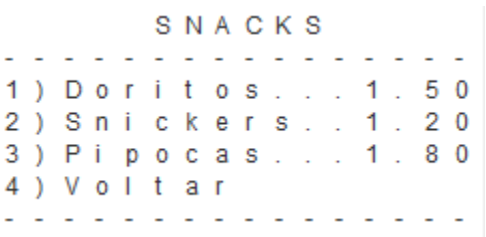
1. PRODUTOS

Escolhendo a opção dos produtos temos então mais um menu para poder escolher entre bebidas e snacks ou até mesmo para cancelar.

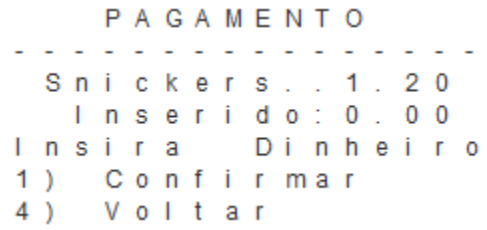


Selecionando uma das opções, seja de bebidas ou de snacks, vai dar uma lista de produtos das mesmas categorias.

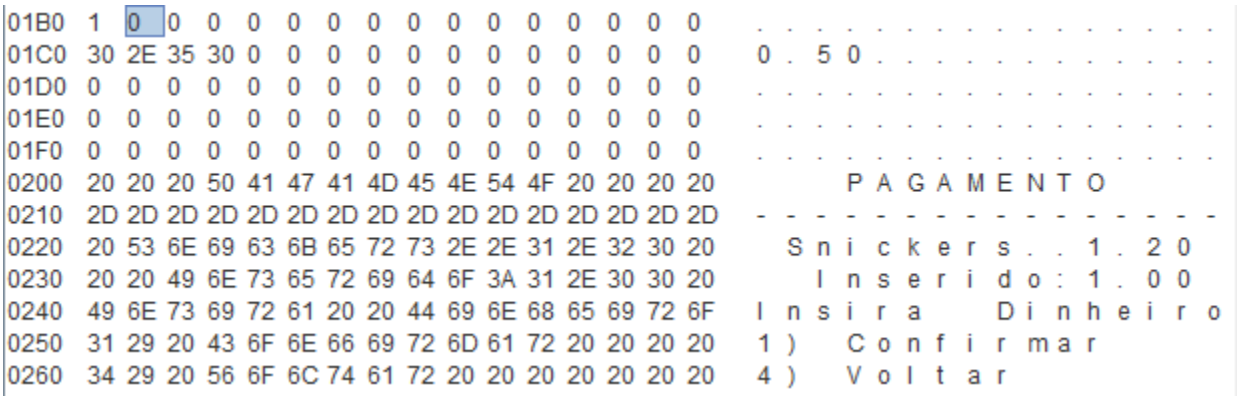




Selecionando um qualquer produto, vou dar o exemplo dos Snickers, ele passará então para o menu de pagamento no qual deve ser inserido as moedas.



É preciso então adicionar moedas para atingir o valor do produto por exemplo.



Neste caso para confirmar, o endereço da opção tem de ser 1 e o endereço do dinheiro tem de ser uma moeda válida, aceitando apenas os valores 5.00, 2.00, 1.00, 0.50, 0.20, 0.10.

Passamos então para o talão pois já chegámos ao dinheiro pretendido, tendo inserido 1.50€. O troco será então 0.30€.

```
- - - - - - - - - - - - - - - - - -
          T A L A O
- - - - - - - - - - - - - - - - - -
S n i c k e r s . . 1 . 2 0
I n s e r i d o . . 1 . 5 0
T r o c o . . . . . 0 . 3 0
- - - - - - - - - - - - - - - - - -
```

2. STOCK

Todos os produtos e dinheiro do stock é inicializado a 10.

Indo para o stock, aproveitando o mesmo exemplo temos que foi inserido uma moeda de 1€ e uma moeda de 0.50€, dará como troco uma moeda de 0.20€ e uma moeda de 0.10€. Como também remover um Snickers do Stock.

Primeiro teremos que inserir a password para entrar no stock.

```
01B0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    . . . . . .
01C0  4A 6E 32 23 0  0  0  0  0  0  0  0  0  0  0  0    J n 2 # . . . . . .
01D0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    . . . . . .
01E0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    . . . . . .
01F0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    . . . . . .
0200  2D 2D 2D 2D 20 53 74 6F 63 6B 20 2D 2D 2D 2D 2D    - - - -   S t o c k   - - - - -
0210  2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D    - - - - - - - - - - - - - - - -
0220  20 20 20 49 6E 74 72 6F 64 75 7A 61 20 20 20 20        I n t r o d u z a
0230  20 20 20 20 50 61 73 73 77 6F 72 64 20 20 20 20          P a s s w o r d
0240  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0250  31 29 20 43 6F 6E 66 69 72 6D 61 72 20 20 20 20    1 )   C o n f i r m a r
0260  34 29 20 56 6F 6C 74 61 72 20 20 20 20 20 20 20    4 )   V o l t a r
```

Passando então para a verificação do Stock temos.

```
- -   S t o c k   1 / 3   - - -
C o c a C o l a . . . . . 1 0
B r i s a . . . . . 1 0
C e r v e j a . . . . . 1 0
D o r i t o s . . . . . 1 0
S n i c k e r s . . . . . 0 9
1 ) S e g u i n t e
- -   S t o c k   3 / 3   - - -
2 0 C e n t . . . . . 0 9
1 0 C e n t . . . . . 0 9


4 ) V o l t a r
```

```
- -   S t o c k   2 / 3   - - -
P i p o c a s . . . . . 1 0
5 E u r o s . . . . . 1 0
2 E u r o s . . . . . 1 0
1 E u r o . . . . . 1 1
5 0 C e n t . . . . . 1 1
1 ) S e g u i n t e
```

Reparamos então que foi tudo removido e adicionado conforme esperado.

CONCLUSÃO

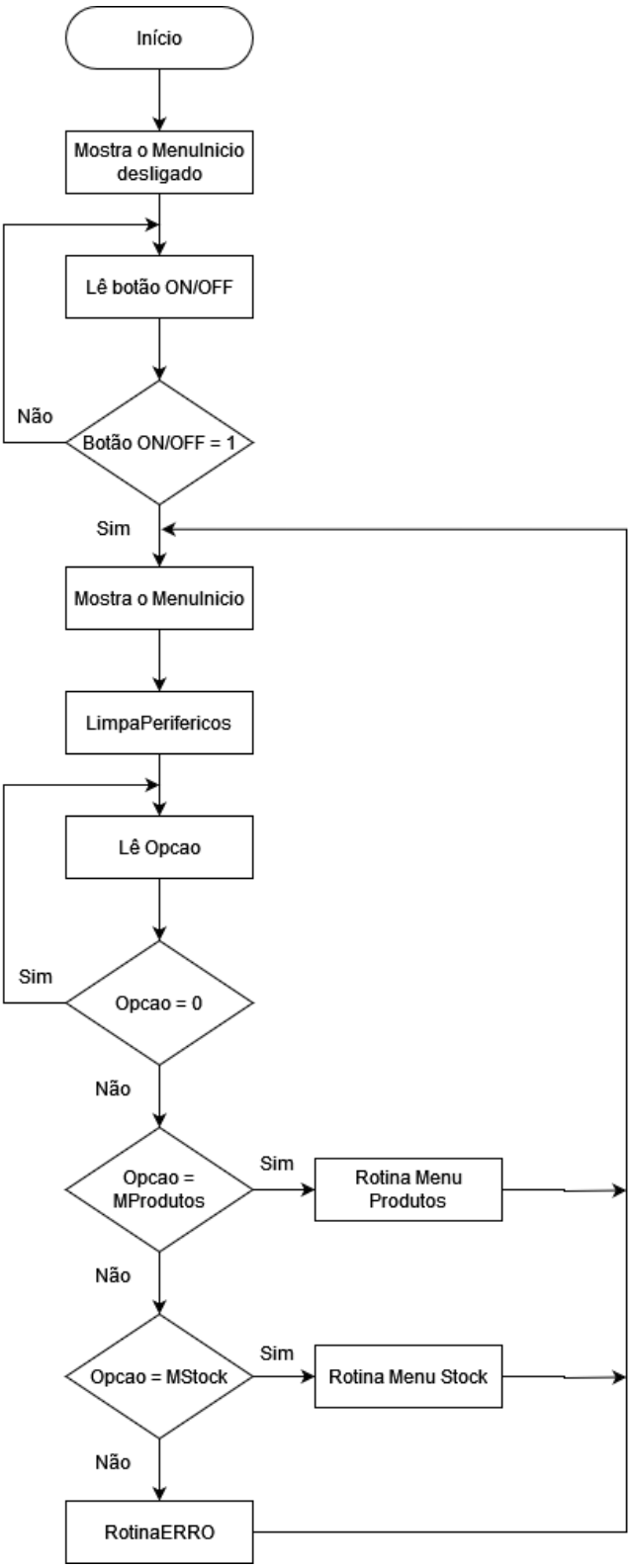
Para resumir, consideramos que alcançamos os objetivos do trabalho ao dividir o problema em partes menores e mais acessíveis. A elaboração prévia de fluxogramas facilitou o desenvolvimento da solução. Embora a linguagem Assembly possa ser mais complexa que uma linguagem de alto nível, ela é mais adequada para programar dispositivos do que o código máquina. De maneira geral, este trabalho foi útil para a compreensão da arquitetura de computadores e seu funcionamento.

BIBLIOGRAFIA

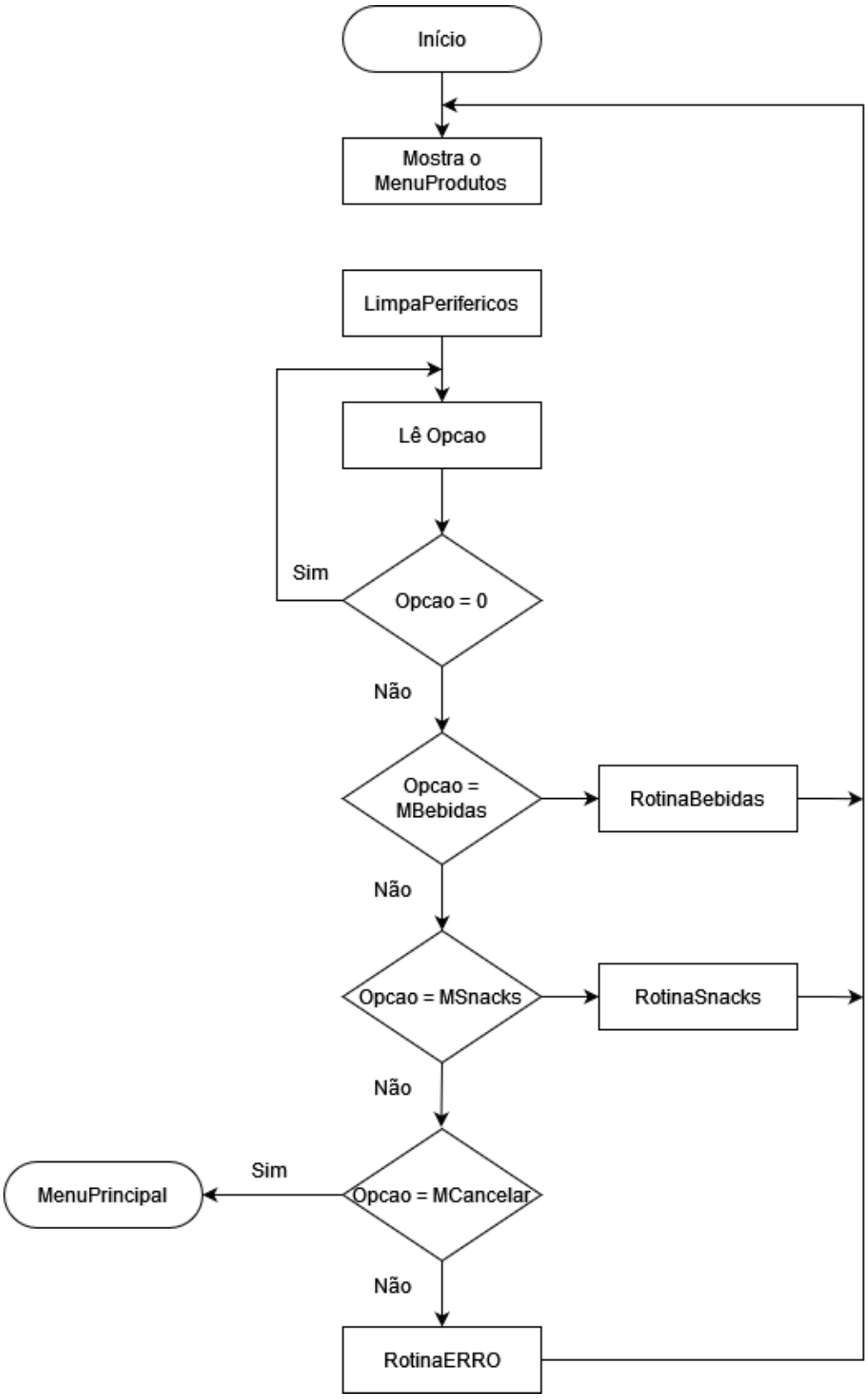
J. Delgado e C. Ribeiro, Arquitetura de Computadores, FCA - Editora de Informática, 2010.

ANEXO A

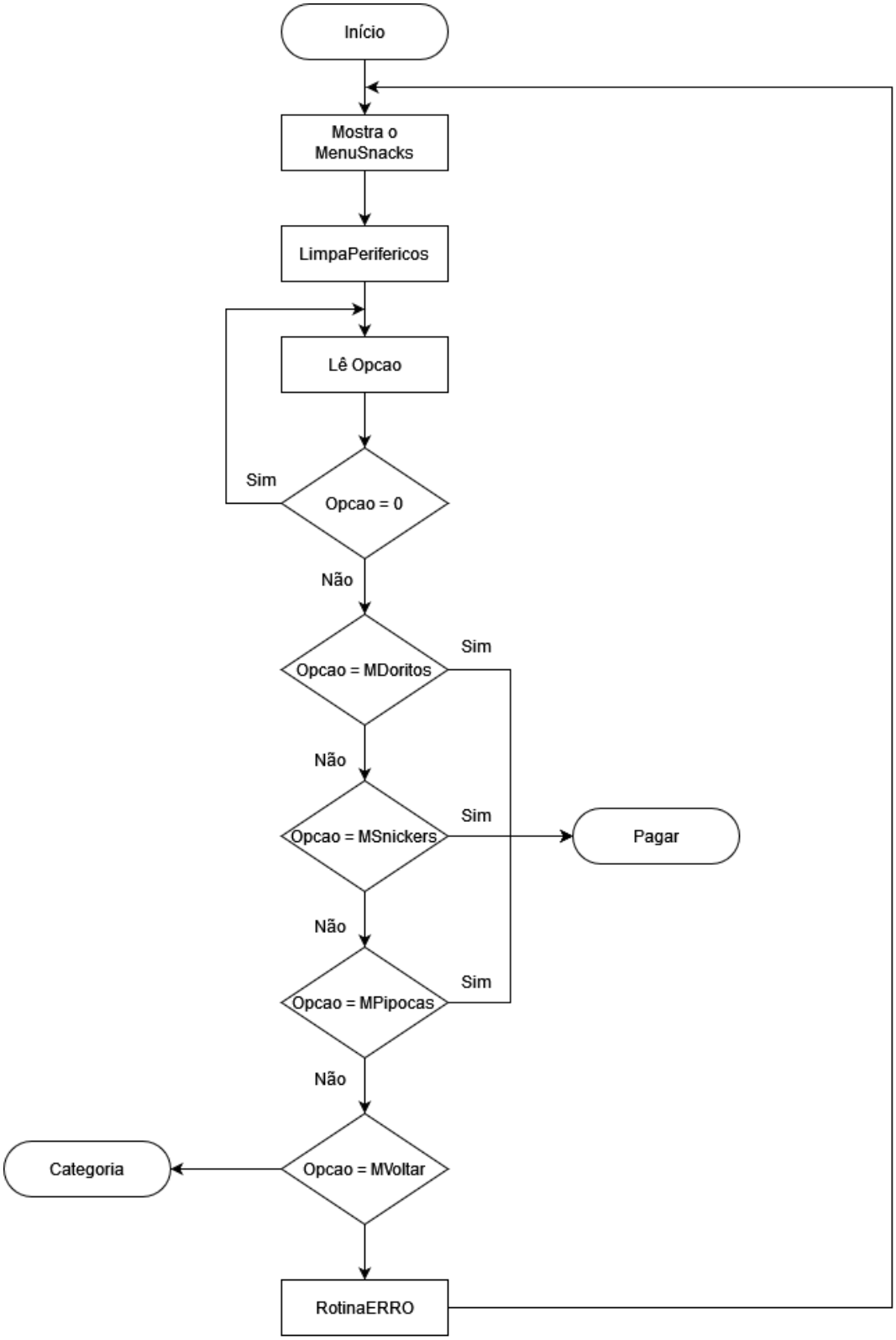
Inicio



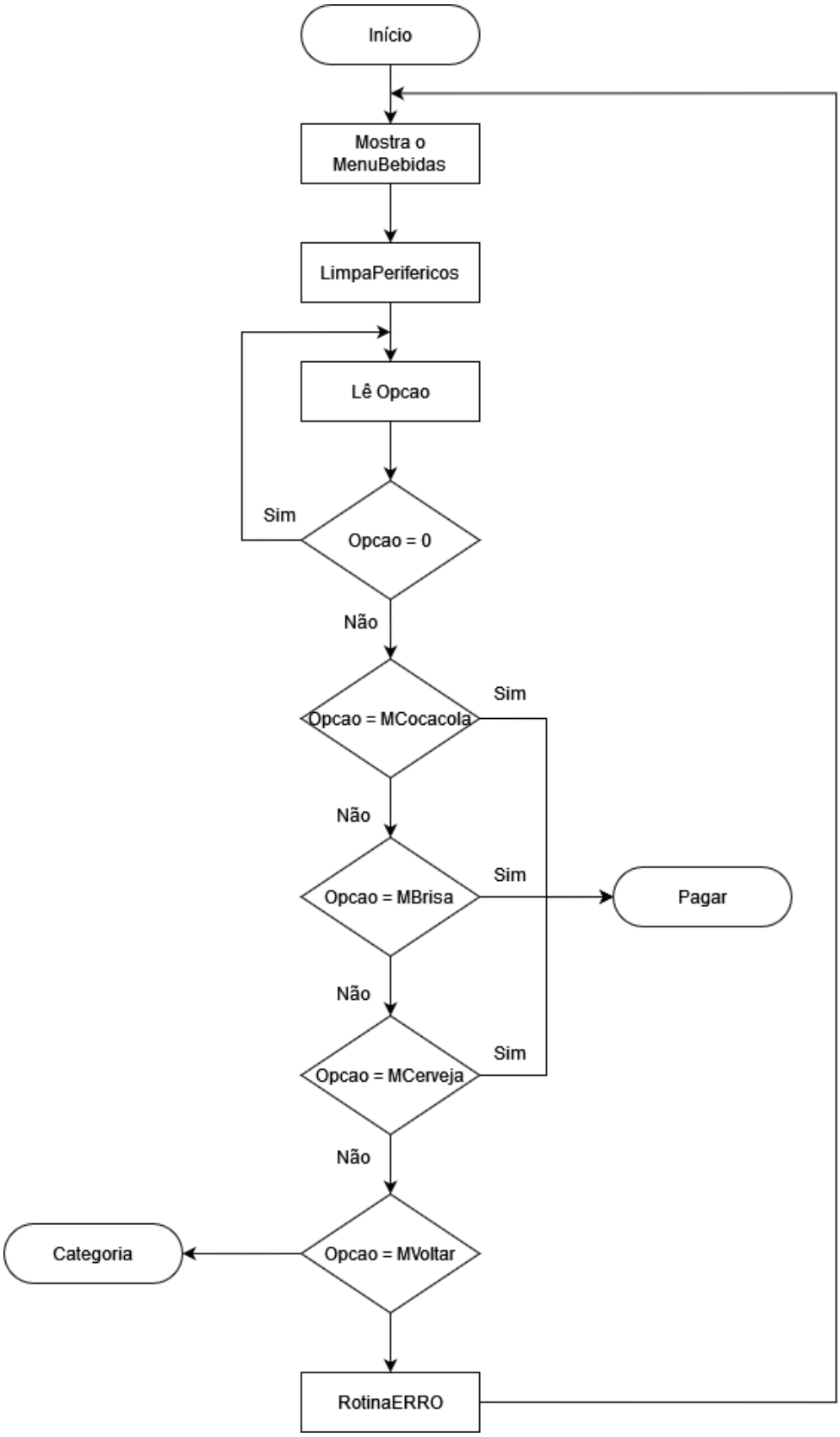
Produtos


Início
Mostra o MenuProdutos
LimpaPerifericos
Lê Opcao
Sim
Opcao = 0
Não
Opcao = MBebidas
RotinaBebidas
Não
Opcao = MSnacks
RotinaSnacks
Não
Sim
MenuPrincipal
Opcao = MCancelar
Não
RotinaERRO

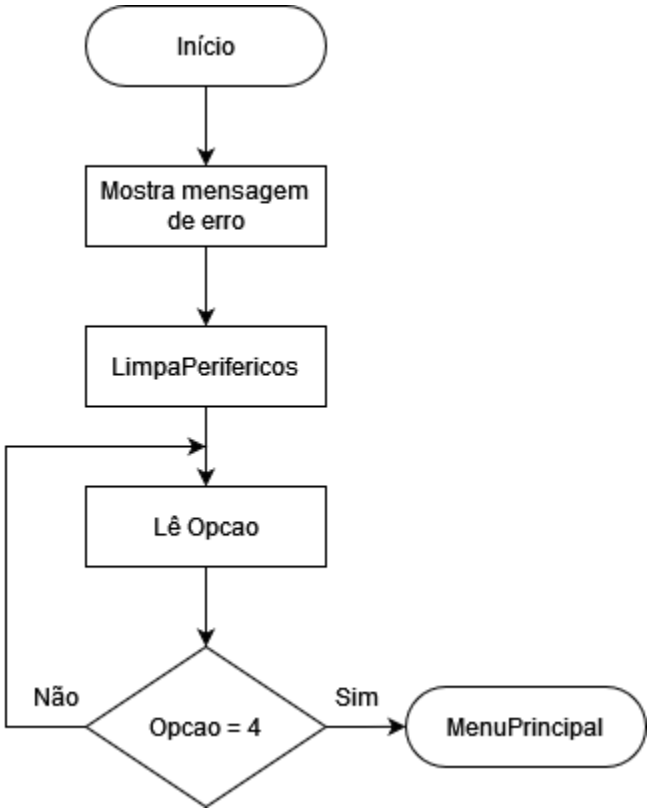
Snacks


Início
Mostra o MenuSnacks
LimpaPerifericos
Lê Opcao
Sim
Opcao = 0
Não
Opcao = MDoritos
Sim
Não
Opcao = MSnickers
Sim
Pagar
Não
Opcao = MPipocas
Sim
Não
Categoria
Opcao = MVoltar
RotinaERRO

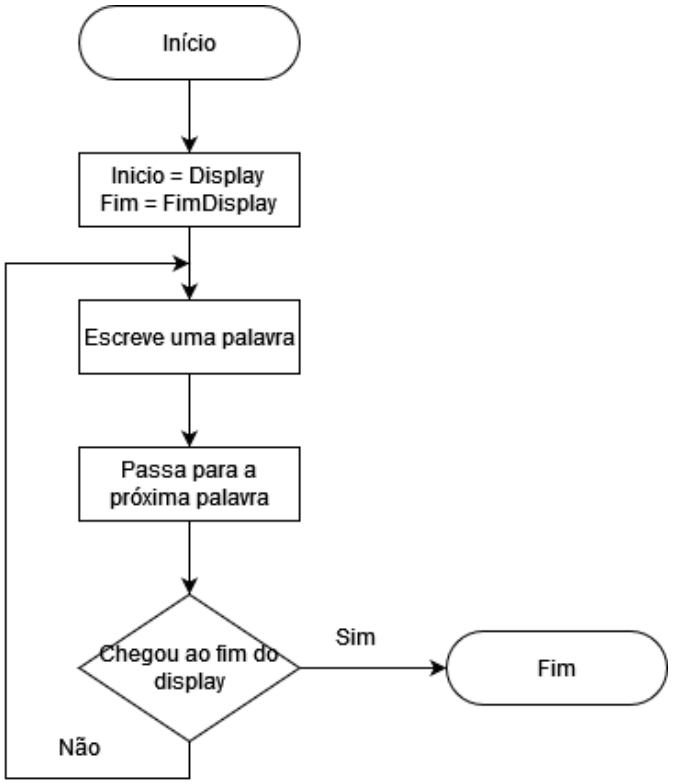
Bebidas


Início
Mostra o MenuBebidas
LimpaPerifericos
Lê Opcao
Opcao = 0
Sim
Não
Opcao = MCocacola
Sim
Não
Opcao = MBrisa
Sim
Pagar
Não
Opcao = MCerveja
Sim
Não
Opcao = MVoltar
Categoria
RotinaERRO

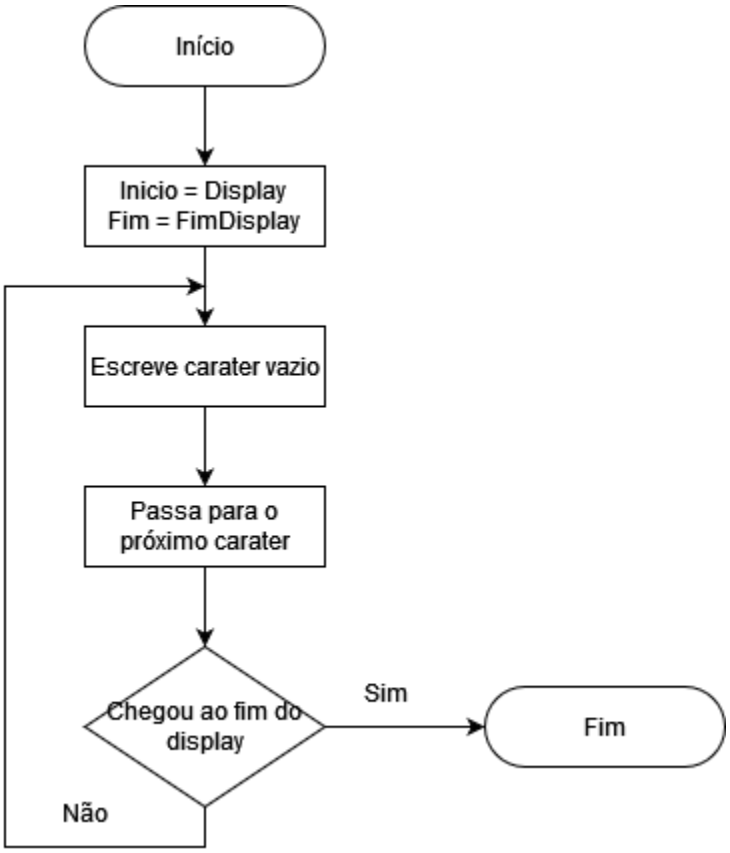
Rotina ERRO



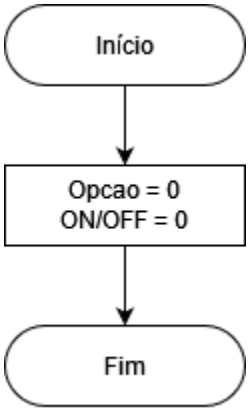
MostraDisplay



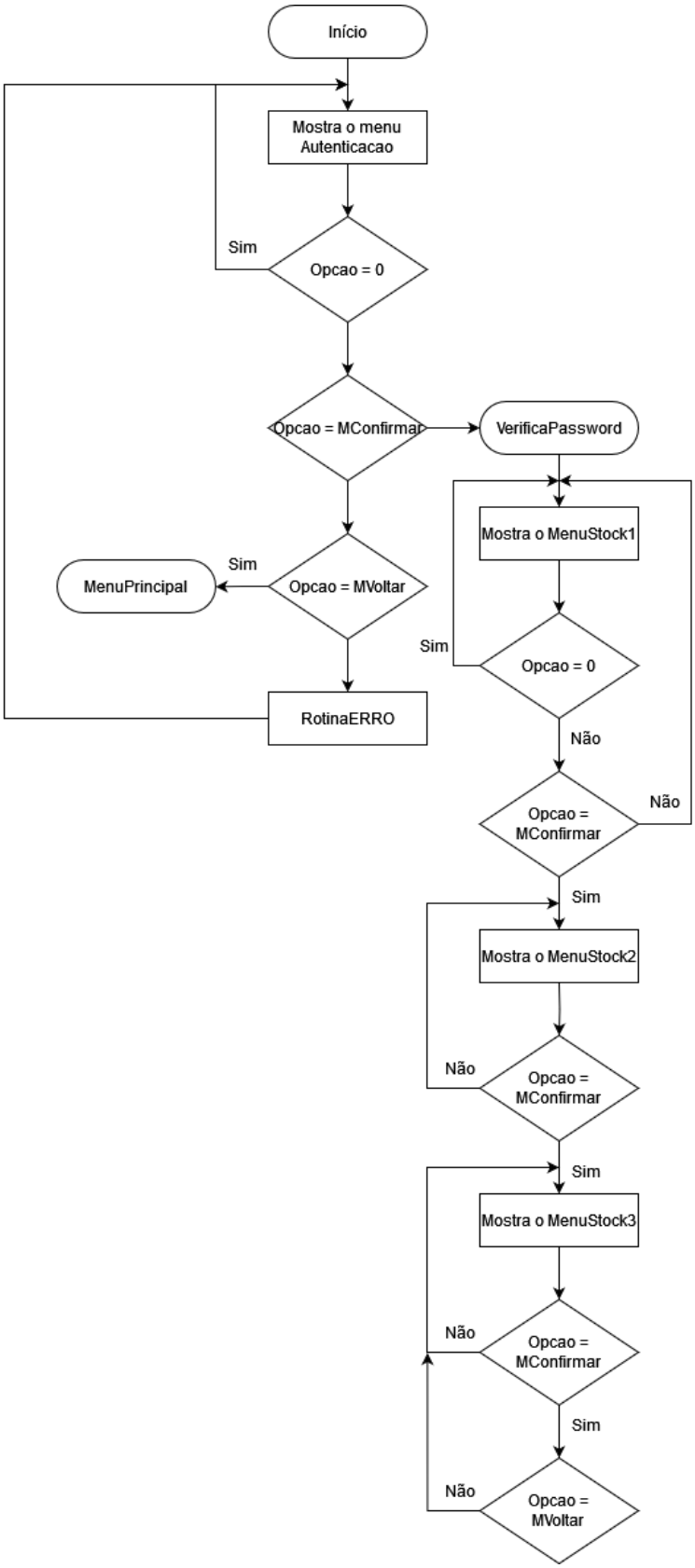
LimpaDisplay



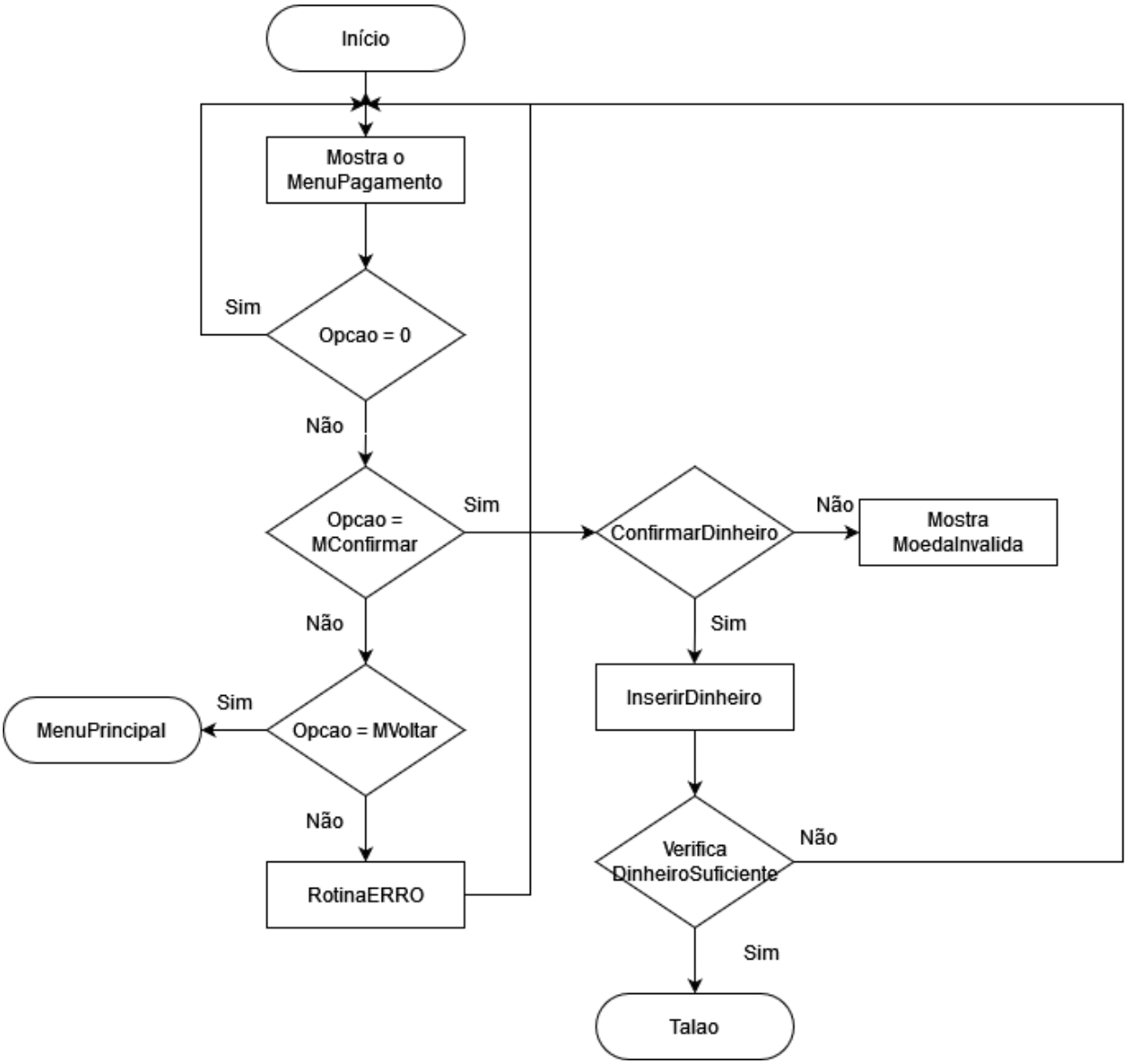
LimpaPerifericos



Stock


Início
Mostra o menu Autenticacao
Sim
Opcao = 0
Opcao = MConfirmar
VerificaPassword
Mostra o MenuStock1
Sim
MenuPrincipal
Opcao = MVoltar
Sim
Opcao = 0
RotinaERRO
Não
Não
Opcao = MConfirmar
Sim
Mostra o MenuStock2
Não
Opcao = MConfirmar
Sim
Mostra o MenuStock3
Não
Opcao = MConfirmar
Sim
Não
Opcao = MVoltar

Pagamento


Início
Mostra o MenuPagamento
Sim
Opcao = 0
Não
Opcao = MConfirmar
Sim
ConfirmarDinheiro
Não
Mostra MoedaInvalida
Não
Sim
InserirDinheiro
Opcao = MVoltar
Sim
MenuPrincipal
Não
RotinaERRO
Verifica DinheiroSuficiente
Não
Sim
Talao

ANEXO B

```
;Endereços
ON_OFF                  EQU 1A0H        ; Botão ON/OFF da máquina
Opcao                   EQU 1B0H        ; Input da opção de cada Menu
PER_EN                  EQU 1C0H        ; Input da Password e do Dinheiro
DinheiroInserido    EQU 23BH        ; Endereço do dinheiro total inserido
;Display
Display             EQU 200H            ; Inicio do Display
Display_End     EQU 26FH            ; Fim do Display
CaraterVazio    EQU 20H             ; Carater para limpar o ecra
;Opcoes Inicio
MProdutos           EQU 1               ; Opção de Produtos
MStock                  EQU 2           ; Opção de Stock
;Opcoes Categoria
MBebidas            EQU 1               ; Opção de Bebidas
MSnacks             EQU 2               ; Opção de Snacks
MCancelar           EQU 7               ; Opção para Cancelar
;Opçoes Bebidas
MCocacola           EQU 1               ; Opção de CocaCola
MBrisa              EQU 2               ; Opção de Brisa
MCerveja            EQU 3               ; Opção de Cerveja
;Opcoes Snacks
MDoritos            EQU 1               ; Opção de Doritos
MSnickers           EQU 2               ; Opção de Snickers
MPipocas            EQU 3               ; Opção de Pipocas
;Opcoes Autenticacao
MConfirmar          EQU 1               ; Opção para Confirmar
MVoltar             EQU 4               ; Opção para Voltar

StackPointer    EQU 6000H
```

```
; R2 é utilizado para guardar o Menu ativo
; R5 é utilizado para guardar o Endereço do Stock
; R7 é utilizado para guardar o Endereço do Dinheiro Inserido
; R9 é utilizado para guardar o Endereço do Dinheiro

PLACE 1000H
ContaStock: STRING "Jn2#"

PLACE 2000H
MenuInicio:
   STRING "----------------"
        STRING "MAQUINA MADEIRA "
        STRING "   BEM-VINDO    "
        STRING "----------------"
        STRING "1) Produtos     "
        STRING "2) Stock        "
        STRING "----------------"

PLACE 2080H
MenuCategoria:
   STRING "----------------"
   STRING "-- Categoria ---"
   STRING "----------------"
   STRING "1) Bebidas      "
   STRING "2) Snacks       "
   STRING "7) Cancelar     "
   STRING "----------------"

PLACE 2100H
Talao:
   STRING "----------------"
   STRING "     TALAO      "
   STRING "----------------"
```

```
    STRING "                "
    STRING "Inserido..X.X0  "
    STRING "Troco.....X.X0  "
    STRING "----------------"

PLACE 2180H
Autenticacao:
        STRING "---- Stock -----"
    STRING "----------------"
    STRING "   Introduza    "
    STRING "    Password    "
    STRING "                "
    STRING "1) Confirmar    "
    STRING "4) Voltar       "

Place 2200H
MenuERRO:
        STRING "    ATENÇÃO     "
        STRING "                "
        STRING "     OPÇÃO      "
        STRING "    ERRADA!     "
        STRING "                "
        STRING "4) Voltar       "
        STRING "                "

PLACE 2280H
MenuPagamento:
        STRING "   PAGAMENTO    "
        STRING "----------------"
        STRING " XXXXXXXXXXXXXXX "
        STRING "  Inserido:0.00 "
        STRING "Insira  Dinheiro"
        STRING "1) Confirmar    "
```

```
        STRING "4) Voltar       "

PLACE 2300H
MenuBebidas:
        STRING "    BEBIDAS     "
        STRING "----------------"
        STRING "1)CocaCola..0.90"
        STRING "2)Brisa.....1.80"
        STRING "3)Cerveja...1.00"
        STRING "4)Voltar        "
        STRING "----------------"

PLACE 2380H
MenuSnacks:
        STRING "     SNACKS     "
        STRING "----------------"
        STRING "1)Doritos...1.50"
        STRING "2)Snickers..1.20"
        STRING "3)Pipocas...1.80"
        STRING "4)Voltar        "
        STRING "----------------"

PLACE 2400H
MenuStock1:
        STRING "-- Stock 1/3 ---"
        STRING "CocaCola......10"
        STRING "Brisa.........10"
        STRING "Cerveja.......10"
        STRING "Doritos.......10"
        STRING "Snickers......10"
        STRING "1)Seguinte      "

PLACE 2480H
```

```
MenuStock2:
        STRING "-- Stock 2/3 ---"
        STRING "Pipocas.....10"
        STRING "5Euros.....10"
        STRING "2Euros.....10"
        STRING "1Euro.....10"
        STRING "50Cent.....10"
        STRING "1)Seguinte      "

PLACE 2500H
MenuStock3:
        STRING "-- Stock 3/3 ---"
        STRING "20Cent.....10"
        STRING "10Cent.....10"
        STRING "              "
        STRING "              "
        STRING "              "
        STRING "4)Voltar       "

PLACE 2580H
MenuFalhaPassword:
        STRING "    ATENÇÃO     "
        STRING "                "
        STRING "    PASSWORD    "
        STRING "     ERRADA!    "
        STRING "                "
        STRING "4) Voltar       "
        STRING "                "

Place 2600H
MenuMoedaInvalida:
        STRING "    ATENÇÃO     "
        STRING "                "
```

```
        STRING "     MOEDA      "
        STRING "   INVÁLIDA!    "
        STRING "                "
        STRING "4) Voltar       "
        STRING "                "

PLACE 0000H
Inicio:
   MOV R0, Principio
   JMP R0

PLACE 3000H
Principio:
   MOV SP,StackPointer
   CALL LimpaDisplay
   CALL LimpaPerifericos
   MOV R0, ON_OFF
Liga:
   MOVB R1,[R0]
   CMP R1,1
   JNE Liga
Ligado:
   MOV R2, MenuInicio
   CALL MostraDisplay
   CALL LimpaPerifericos
Le_Opcao:
   MOV R0,Opcao
   MOVB R1, [R0]
   CMP R1,0
   JEQ Le_Opcao
   CMP R1, MProdutos
   JEQ OCategoria
   CMP R1, MStock
```

```
        JEQ OStock
    CALL RotinaERRO
    JMP Ligado

;-------------------------------------------------
;       Menu Categoria
;-------------------------------------------------
OCategoria:
        MOV R2, MenuCategoria
        CALL MostraDisplay
        CALL LimpaPerifericos
CicloCategoria:
        MOV R0, Opcao
        MOVB R1, [R0]                   ; Guarda no Registo 1 o valor no
Endereço da Opção guardado no Registo 0
        CMP R1, 0
        JEQ CicloCategoria              ; Caso a opção continue igual a 0
continua o ciclo
        CMP R1, MBebidas
        JEQ OBebidas                    ; Caso a opção seja a de Bebidas salta
para a Rotina Bebidas
        CMP R1, MSnacks
        JEQ OSnacks                     ; Caso a opção seja a de Snacks
salta para a Rotina Snacks
        CMP R1, MCancelar
        JEQ FimCategoria
        CALL RotinaERRO
        JMP OCategoria
FimCategoria:
        JMP Ligado

;-------------------------------------------------
;       Menu Bebidas
;-------------------------------------------------
```

```
OBebidas:
        MOV R2, MenuBebidas
        CALL MostraDisplay
        CALL LimpaPerifericos
CicloBebidas:
        MOV R0, Opcao
        MOV R5, 2410H                   ; Guarda no Registo 5 o Endereço do Stock da opção CocaCola
        MOVB R1, [R0]                   ; Guarda no Registo 1 o valor no Endereço da Opção guardado no Registo 0
        CMP R1, 0
        JEQ CicloBebidas                ; Caso a opção continue igual a 0 continua o ciclo
        CMP R1, MCocacola
        JEQ Pagar                       ; Caso a opção seja a de CocaCola do Stock salta para a Rotina Pagar
        MOV R3, 16
        ADD R5, R3                      ; Guarda no Registo 5 o Endereço do Stock da opção Brisa incrementando 16 ao endereço
        CMP R1, MBrisa
        JEQ Pagar                       ; Caso a opção seja a de Brisa salta para a Rotina Pagar
        ADD R5, R3                      ; Guarda no Registo 5 o Endereço do Stock da opção Cerveja incrementando 16 ao endereço
        CMP R1, MCerveja
        JEQ Pagar                       ; Caso a opção seja a de Cerveja salta para a Rotina Pagar
        CMP R1, MVoltar
        JEQ OCategoria
        CALL RotinaERRO
        JMP OBebidas


;-------------------------------------------------
;       Menu Snacks
;-------------------------------------------------
```

```
OSnacks:
        MOV R2, MenuSnacks
        CALL MostraDisplay
        CALL LimpaPerifericos
CicloSnacks:
        MOV R0, Opcao
        MOV R5, 2440H                   ; Guarda no Registo 5 o
Endereço do Stock da opção CocaCola
        MOVB R1, [R0]
        CMP R1, 0
        JEQ CicloSnacks                 ; Caso a opção continue igual a 0
continua o ciclo
        CMP R1, MDoritos
        JEQ Pagar                       ; Caso a opção seja a de Doritos
salta para a Rotina Pagar
        MOV R3, 16
        ADD R5, R3                      ; Guarda no Registo 5 o
Endereço do Stock da opção Snickers incrementando 16 ao endereço
        CMP R1, MSnickers
        JEQ Pagar                       ; Caso a opção seja a de Snickers
salta para a Rotina Pagar
        MOV R3, 64
        ADD R5, R3                      ; Guarda no Registo 5 o
Endereço do Stock da opção Pipocas incrementando 64 ao endereço
        CMP R1, MPipocas
        JEQ Pagar                       ; Caso a opção seja a de Pipocas
salta para a Rotina Pagar
        CMP R1, MVoltar
        JEQ OCategoria
        CALL RotinaERRO
        JMP OSnacks


Pagar:
        MOV R3, 16
        ADD R1, 1
```

```
        MUL R3, R1
        ADD R3, R2                          ; Guarda no Registo 3 o Endereço do Produto da opção escolhida
        JMP OPagamento
;-------------------------------------------------
;       Menu Stock - Aqui é esperada a inserção da password
;-------------------------------------------------
OStock:
        MOV R2, Autenticacao
        CALL MostraDisplay
        CALL LimpaPerifericos
CicloStock:
        MOV R0, Opcao
        MOVB R1, [R0]
        CMP R1, 0                           ; Caso a opção continue igual a 0 continua o ciclo
        JEQ CicloStock
        CMP R1, MConfirmar
        JEQ ConfimarStock                   ; Caso o utilizador tenha inserido a pass e confirmado ele vai verificar se a password está correta
        CMP R1, MVoltar
        JEQ FimStock                        ; Volta para o menu Inicial
        CALL RotinaERRO
        JMP OStock

ConfimarStock:
        JMP VerificaPassword

FimStock:
        JMP Ligado

;-------------------------------------------------
;       Menus StockList - Aqui é mostrado o Stock da máquina nas várias páginas
;-------------------------------------------------
```

```
OStockList1:
        MOV R2, MenuStock1
        CALL MostraDisplay
        CALL LimpaPerifericos
CicloStockList1:
        MOV R0, Opcao
        MOVB R1, [R0]
        CMP R1, 0
        JEQ CicloStockList1                    ; Caso a opção continue igual a 0
continua o ciclo
        CMP R1, MConfirmar
        JEQ OStockList2
        JMP CicloStockList1
OStockList2:
        MOV R2, MenuStock2
        CALL MostraDisplay
        CALL LimpaPerifericos
CicloStockList2:
        MOV R0, Opcao
        MOVB R1, [R0]
        CMP R1, 0
        JEQ CicloStockList2                    ; Caso a opção continue igual a 0
continua o ciclo
        CMP R1, MConfirmar
        JEQ OStockList3
        JMP CicloStockList2
OStockList3:
        MOV R2, MenuStock3
        CALL MostraDisplay
        CALL LimpaPerifericos
CicloStockList3:
        MOV R0, Opcao
        MOVB R1, [R0]
```

```
        CMP R1, 0
        JEQ CicloStockList3                     ; Caso a opção continue igual a 0 continua o ciclo
        CMP R1, MVoltar
        JEQ OStock
        JMP CicloStockList3



;-------------------------------------------------
;       Menu Pagamento - Apresenta o Menu Pagamento no qual é usado para inserir o dinheiro para efetuar a compra
;-------------------------------------------------
OPagamento:
        MOV R7, R3                              ; Guarda em R6 o valor do Endereço da opção escolhida
        MOV R2, MenuPagamento
        CALL MostraDisplay
        CALL LimpaPerifericos
        ADD R7, 2                               ; Incrementa dois para retirar o número e o parenteses, por exemplo "2)"
        MOV R4, Display                         ; Guarda em R4 o Endereço do Display
        MOV R8, 33
        ADD R4, R8                              ; Incrementa 33 ao valor para poder começar a escrever o produto no display do pagamento
        MOV R1, 0                               ; Inicializa o Registo 1 a 0
        MOV R8, 13
CicloProduto:                                   ; Impressão do produto no display do pagamento
        MOVB R10, [R7]                          ; Guarda em R10 o valor no Endereço guardado em R7
        MOVB [R4], R10                          ; Guarda no valor do endereço R4 o valor de R10
        CMP R1, R8                              ; Verifica se chegou ao fim da impressão
```

```
        JEQ CicloPagamento                  ; Se for igual passa para o pagamento do produto
        ADD R1, 1                           ; Incrementa o contador R1
        ADD R7, 1                           ; Incrementa a posição R7
        ADD R4, 1                           ; Incrementa a posição R4
        JMP CicloProduto
CicloPagamento:
        MOV R0, Opcao
        MOVB R1, [R0]
        MOV R9, PER_EN                      ; Guarda o valor do endereço do dinheiro inserido no formato X.XX
        MOVB R6, [R9]                       ; Guarda o valor do endereço R9
        CMP R1, 0
        JEQ CicloPagamento                  ; Caso a opção continue igual a 0 continua o ciclo
        CMP R1, MConfirmar
        JEQ ConfirmarDinheiro
        CMP R1, MVoltar
        JEQ FimPagamento
        CALL RotinaERRO
        JMP OPagamento

FimPagamento:
        JMP OCategoria

;-------------------------------------------------
;       Mostra Display - Mostra o display que está guardado no Registo 2
;-------------------------------------------------

MostraDisplay:
   PUSH R0
   PUSH R1
        PUSH R2
   PUSH R3
```

```
    MOV R0, Display
    MOV R1, Display_End
Ciclo:
    MOV R3,[R2]
    MOV [R0], R3
    ADD R2,2
    ADD R0,2
    CMP R0,R1
    JLE Ciclo
    POP R3
    POP R2
        POP R1
    POP R0
    RET

;--------------------------------------------------
;       Limpa Perifericos - Limpa os perifericos de ligar/desligar a máquina e o da opção
;--------------------------------------------------
LimpaPerifericos:
    PUSH R0
    PUSH R1
    PUSH R3
    MOV R0, ON_OFF
    MOV R1, Opcao
    MOV R3, 0
    MOVB [R0], R3
    MOVB [R1], R3
    POP R3
    POP R1
    POP R0
    RET

;--------------------------------------------------
```

```
;       Limpa Display - Limpa o Display atual para carateres vazios
;-------------------------------------------------
LimpaDisplay:
    PUSH R0
    PUSH R1
    PUSH R3
    MOV R0, Display
    MOV R1, Display_End
CicloLimpaDisplay:
    MOV R3, CaraterVazio
    MOVB [R0], R3
    ADD R0,1
    CMP R0,R1
    JLE CicloLimpaDisplay
    POP R3
    POP R1
    POP R0
    RET

;-------------------------------------------------
;       Rotina Erro - Aprensenta um erro caso algo esta mal
;-------------------------------------------------
RotinaERRO:
    PUSH R0
    PUSH R1
    PUSH R2
    MOV R2, MenuERRO
    CALL MostraDisplay
    CALL LimpaPerifericos
    MOV R0, Opcao
ERRO:
    MOVB R1,[R0]
    CMP R1, 4
```

```
        JEQ FimErro
    JMP ERRO
FimErro:
        POP R2
    POP R1
    POP R0
        JMP Ligado



;--------------------------------------------------
;       Verifica Password - Aqui é verificada se a password introduzida para entrar no stock é válida
;--------------------------------------------------


VerificaPassword:
        MOV R8, 0                       ; Contador de carateres neste caso esta password tem 4 carateres
        MOV R6, PER_EN                  ; Guarda em R6 o endereço da password inserida pelo utilizador
        MOV R10, ContaStock             ; Guarda em R10 o endereço da password necessária para entrar no Stock
CicloPassword:
        MOVB R4, [R6]                   ; Guarda em R4 o valor do endereço R6
        MOVB R1, [R10]                  ; Guarda em R1 o valor do endereço R10
        CMP R4, R1                      ; Verifica se os carateres são iguais
        JNE ErroPassword
        ADD R6, 1                       ; Incrementa em 1 para verificar o proximo carater da password inserida pelo utilizador
        ADD R10, 1
        ADD R8, 1
        CMP R8, 4                       ; Verifica se já verificou os 4 carateres da password
        JEQ FimPassword
```

```
        JMP CicloPassword
FimPassword:
        CALL LimparPER                          ;
        CALL LimpaPerifericos                   ; Entra nos menus da StockList caso a password seja a correta
        JMP OStockList1                         ;
ErroPassword:
        MOV R2, MenuFalhaPassword               ;
        CALL MostraDisplay                      ; Se não corresponderem as passwords ele vai para o Menu de Password Inválida
        CALL LimparPER                          ;
        CALL LimpaPerifericos                   ;
CicloErroPassword:
        MOV R0, Opcao
        MOVB R1, [R0]
        CMP R1, 0
        JEQ CicloErroPassword
        CMP R1, 4
        JEQ FimErroPassword
        JMP CicloErroPassword
FimErroPassword:
        JMP OStock
;-------------------------------------------------
;       Verificar Pagamento
; 1º Verifica se a moeda ou nota inserida é válida
; 2º Adiciona a moeda ou nota ao Stock
; 3º Adiciona o valor da moeda ao valor já inserido
; 4º Verifica se o dinheiro é suficiente para o produto
;-------------------------------------------------

ConfirmarDinheiro:
        PUSH R5
        PUSH R9
```

```
        ADD R9, 3
        MOVB R8, [R9]                   ; Passa para o útlimo valor do
dinheiro que foi introduzido
        MOV R10, 30H
        CMP R8, R10                     ; Verifica se a moeda inserida
tem um 0 no último valor ou seja X.X0
        JNE OMoedaInvalida
        SUB R9, 3
        MOVB R8, [R9]
        CMP R8, R10                     ; Verifica se a moeda tem 0
euros, ou seja, 0.X0
        JEQ ParaCentimos
        MOV R10, 31H
        MOV R5, 24C0H
        CMP R8, R10                     ; Verifica se a moeda tem 1
euros, ou seja, 1.X0
        JEQ CentimosEuros
        MOV R10, 32H
        MOV R5, 24B0H
        CMP R8, R10                     ; Verifica se a moeda tem 2
euros, ou seja, 2.X0
        JEQ CentimosEuros
        MOV R10, 35H
        MOV R5, 24A0H
        CMP R8, R10                     ; Verifica se a nota tem 5 euros,
ou seja, 5.X0
        JEQ CentimosEuros
        JMP OMoedaInvalida
ParaCentimos:
        ADD R9, 1
        MOVB R8, [R9]
        MOV R10, 2EH
        CMP R8, R10                     ; Verifica se tem um ponto entre
os euros e os cêntimos se não tiver dá erro
        JNE OMoedaInvalida
```

```
        ADD R9, 1
        MOVB R8, [R9]
        MOV R10, 30H
        CMP R8, R10                     ; Verifica se a moeda tem o valor
de 0.00, se tiver dá erro
        JEQ OMoedaInvalida
        MOV R10, 31H
        MOV R5, 2520H
        CMP R8, R10                     ; Verifica se a moeda tem o valor
de 0.10
        JEQ FimVerificar
        MOV R10, 32H
        MOV R5, 2510H
        CMP R8, R10                     ; Verifica se a moeda tem o valor
de 0.20
        JEQ FimVerificar
        MOV R10, 35H
        MOV R5, 24D0H
        CMP R8, R10                     ; Verifica se a moeda tem o valor
de 0.50
        JEQ FimVerificar
        JMP OMoedaInvalida
CentimosEuros:
        ADD R9, 2
        MOVB R8, [R9]
        MOV R10, 30H
        CMP R8, R10                     ; Verifica se os euros, ou seja,
1.00, 2.00 ou 5.00
        JNE OMoedaInvalida
        JMP FimVerificar

OMoedaInvalida:                         ; Mostra o menu de Moeda
Inválida
        POP R9
        POP R5
```

```
        MOV R2, MenuMoedaInvalida
        CALL MostraDisplay
        CALL LimpaPerifericos
CicloMoedaInvalida:
        MOV R0, Opcao
        MOVB R1, [R0]
        CMP R1, 0
        JEQ CicloMoedaInvalida
        CMP R1, 4
        JEQ FimMoedaInvalida
        JMP CicloMoedaInvalida
FimMoedaInvalida:
        JMP OCategoria
FimVerificar:
        CALL AdicionarAoStock          ; Adiciona a moeda ao stock através do endereço guardado no Registo 5
        POP R9
        POP R5
        JMP InserirDinheiro


VerificarDinheiroSuficiente:
        MOV R7, DinheiroInserido       ; Guarda em R7 o valor do dinheiro total já inserido
        MOV R9, 22BH                   ; Guarda em R9 o endereço que contém valor do produto
        MOVB R4, [R7]                  ; Guarda em R4 o valor do endereço R7
        MOVB R6, [R9]                  ; Guarda em R6 o valor do endereço R9
        CMP R4, R6
        JEQ VerificarCentimos          ; Se os euros forem iguais ele vai comparar os cêntimos
        JLT DinheiroInsuficiente       ; Se os euros forem menores ele espera por mais moedas
```

```
                                                    ; Se os euros forem maiores ele remove o produto do stock e imprime o talao
DinheiroSuficiente:
        CALL RemoverDoStock
        JMP OTalao

VerificarCentimos:
        PUSH R7
        PUSH R9
        ADD R7, 2
        ADD R9, 2
        MOVB R4, [R7]                               ; Guarda em R4 o valor do endereço R7
        MOVB R6, [R9]                               ; Guarda em R6 o valor do endereço R9
        POP R9
        POP R7
        CMP R4, R6
        JGE DinheiroSuficiente                      ; Se os cêntimos forem maiores ou iguais o dinheiro é suficiente se não ele espera por mais moedas voltando ao ciclo

DinheiroInsuficiente:
        JMP CicloPagamento

InserirDinheiro:
        MOV R7, DinheiroInserido                    ; Guarda em R7 o endereço do dinheiro total inserido até agora
        MOV R9, PER_EN                              ; Guarda em R6 o endereço do Dinheiro inserido pelo utilizador
        MOVB R4, [R7]                               ; Guarda em R4 o valor do endereço R7
        MOVB R6, [R9]                               ; Guarda em R6 o valor do endereço R9
        MOV R8, 30H                                 ; Guarda em R8 o hexadecimal de 0
```

```
        SUB R6, R8                                  ; Subtrai para fazer as contas, por
exemplo 31H-30H = 1
        ADD R4, R6                                  ; Adiciona o valor anteriormente
calculado ao valor dos euros do dinheiro total inserido
        MOVB [R7], R4                               ; Regista no endereço de R7 o
valor adicionado dos euros
        ADD R7, 2                                   ; Incrementa o endereço em dois
para o valor dos centimos
        ADD R9, 2                                   ; Incrementa o endereço em dois
para o valor dos centimos
        MOVB R4, [R7]                               ; Guarda em R4 o valor dos
cêntimos
        MOVB R6, [R9]                               ; Guarda em R6 o valor dos
cêntimos
        SUB R6, R8                                  ; Subtrai para fazer as contas, por
exemplo 31H-30H = 1
        ADD R4, R6                                  ; Adiciona o valor anteriormente
calculado ao valor dos cêntimos do dinheiro total inserido
        MOVB [R7], R4                               ; Regista no endereço de R7 o
valor adicionado dos cêntimos
        SUB R9, 2                                   ; Volta à posição dos euros
        SUB R7, 2                                   ; Volta à posição dos euros
        CALL LimparPER
        CALL LimpaPerifericos
        JMP VerificarDinheiroSuficiente             ; Verifica se agora o dinheiro é suficiente
;--------------------------------------------------
;       Limpa PER_EN - Limpa o PER_EN de 4 carateres
;--------------------------------------------------
LimparPER:
        PUSH R4
        PUSH R7
        MOV R7, PER_EN
        MOV R4, 0
        MOVB [R7], R4
        ADD R7, 1
        MOVB [R7], R4
```

```
        ADD R7, 1
        MOVB [R7], R4
        ADD R7, 1
        MOVB [R7], R4
        POP R7
        POP R4
        RET


;-------------------------------------------------
;       Talao - Menu Talao mostrado no final da compra
;-------------------------------------------------
OTalao:
        MOVB R4, [R7]                           ; Guarda em R4 o valor dos euros
do dinheiro total inserido
        ADD R7, 2
        MOVB R1, [R7]                           ; Guarda em R4 o valor dos
centimos do dinheiro total inserido
        MOV R2, Talao
        CALL MostraDisplay
        CALL LimpaPerifericos
        MOV R10, 24AH
        ;**************************************
        MOVB [R10], R4                          ; Transfere o valor inserido para
o talão
        ADD R10, 2                              ;
        MOVB [R10], R1
        ;**************************************
        MOV R4, 230H                        ; Guarda em R4 o endereço no display
para inserir o nome e valor do produto escolhido
        MOV R8, 13
        MOV R1, 0
        MOV R7, R3
        ADD R7, 2
CicloProdutoTalao:                          ; Imprimir o produto seguindo o mesmo
raciocinio que no menu pagamento
```

```
        MOVB R10, [R7]
        MOVB [R4], R10
        CMP R1, R8
        JEQ Troco                       ; Mal o produto esteja impresso
ele vai calcular o troco
        ADD R1, 1
        ADD R7, 1
        ADD R4, 1
        JMP CicloProdutoTalao
CicloTalao:
        MOV R0, Opcao
        MOVB R1, [R0]
        CMP R1, 4
        JEQ FimTalao
        JMP CicloTalao
FimTalao:
        JMP Ligado


Troco:
        MOV R9, 23CH                    ; Endereço do dinheiro do
produto escolhido
        MOV R10, 24CH                   ; Endereço do dinheiro inserido
        MOV R8, 25CH                    ; Endereço onde estará o troco
        MOVB R4, [R9]                   ; Guarda em R4 o valor dos
centimos onde está o endereço
        MOVB R6, [R10]                  ; Guarda em R6 o valor dos
centimos onde está o endereço
        CMP R6, R4
        JLT ExcessaoTroco               ; Verifica se R6 é menor que R4, se for
menor ele vai para a excessão
        SUB R6, R4                      ; Se não for menor ele subtrai um
com o outro dando por exemplo 31-30 = 1
        MOV R7, 30H                     ; Guarda o valor de 30 em R7
        ADD R6, R7                      ; Adiciona 30 para termos o valor
de 1 no fim ou seja 1+30 = 31
```

```
        MOVB [R8], R6                   ; Insere o valor nos centimos do
troco
        SUB R9, 2                       ;
        SUB R10, 2                      ; Decrementa 2 no endereço para
obter o endereço dos euros
        SUB R8, 2                       ;
        MOVB R4, [R9]                   ; Guarda em R4 o valor dos euros
onde está o endereço
        MOVB R6, [R10]                  ; Guarda em R6 o valor dos euros
onde está o endereço
        SUB R6, R4                      ;
        ADD R6, R7                      ; Mesmo processo que acima
referido
        MOVB [R8], R6                   ;
        JMP TrocoDoStock


ExcessaoTroco:
        MOV R1, 10
        ADD R6, R1                      ; Adiciona 10 em R1
        SUB R6, R4                      ; Subtrai o valor pelo do produto,
por exemplo, 40-32 = 8
        MOV R7, 30H                     ; Guarda o valor de 30 em R7
        ADD R6, R7                      ; Adiciona 30 para termos o valor
de 1 no fim ou seja 8+30 = 38
        MOVB [R8], R6                   ; Insere o valor nos centimos do
troco
        SUB R9, 2                       ;
        SUB R10, 2                      ; Decrementa 2 no endereço para
obter o endereço dos euros
        SUB R8, 2                       ;
        MOVB R4, [R9]                   ; Guarda em R4 o valor dos euros
onde está o endereço
        MOVB R6, [R10]                  ; Guarda em R6 o valor dos euros
onde está o endereço
        ADD R4, 1                       ; Adiciona 1 euro ao valor do
produto
```

```
        SUB R6, R4                      ;
        ADD R6, R7                      ; Mesmo processo que acima referido
        MOVB [R8], R6                   ;
        JMP TrocoDoStock


;-------------------------------------------------
;       Remover e Adicionar do Stock
;-------------------------------------------------
RemoverDoStock:
        PUSH R1
        PUSH R3
        PUSH R4
        PUSH R6
        PUSH R7
        PUSH R8
        PUSH R9
        PUSH R10
        MOV R10, R5                     ;
        MOV R8, 15                      ; R10 fica com o endereço do ultimo número da quantidade de um produto ou moeda
        ADD R10, R8                     ;
        MOV R3, 1
        MOV R4, 0                       ; Atribui 0 a R4
        MOV R1, 0                       ; Atribui 0 a R1, R1 é um contador para parar o ciclo seguinte
        MOV R8, 30H
CicloRemover:
        MOVB R9, [R10]
        SUB R9, R8
        MUL R9, R3
        ADD R4, R9
        MOV R7, 10
```

```
        MUL R3, R7                  ; Ao andar para a esquerda no
endereço multiplica por 10
        SUB R10, 1                  ; R4 fica então com a quantidade
de um produto
        ADD R1, 1
        CMP R1, 2
        JLT CicloRemover
        SUB R4, 1                   ; Sendo o Stock a 10, o stock
agora teria que ser 9 pois vamos remover
        MOV R9, R4
        MOV R6, R4
        DIV R9, R7                  ; Obtém o valor dos euros da
subtração do valor, 9/10 = 0
        MOD R6, R7                  ; Obtém o valor dos cêntimos da
subtração do valor, 9%10 = 9
        ADD R10, 1                  ; Incrementa o valor em 1 o
endereço
        ADD R9, R8                  ; Adiciona 30 para obter o valor
correto no display em hexadecimal
        ADD R6, R8                  ; Adiciona 30 para obter o valor
correto no display em hexadecimal
        MOVB [R10], R9              ; Transfere o valor dos euros para
o endereço de R10
        ADD R10, 1
        MOVB [R10], R6              ; Transfere o valor dos centimos
para o endereço de R10
        POP R10
        POP R9
        POP R8
        POP R7
        POP R6
        POP R4
        POP R3
        POP R1
        RET
```

```
AdicionarAoStock:                    ; Mesmo processo que o remover apenas altera a linha comentada
        PUSH R1
        PUSH R3
        PUSH R4
        PUSH R6
        PUSH R7
        PUSH R8
        PUSH R9
        PUSH R10
        MOV R10, R5
        MOV R8, 15
        ADD R10, R8
        MOV R3, 1
        MOV R4, 0
        MOV R1, 0
        MOV R8, 30H
CicloAdicionar:
        MOVB R9, [R10]
        SUB R9, R8
        MUL R9, R3
        ADD R4, R9
        MOV R7, 10
        MUL R3, R7
        SUB R10, 1
        ADD R1, 1
        CMP R1, 2
        JLT CicloAdicionar
        ADD R4, 1                    ; Sendo o Stock a 10, o stock agora teria que ser 11 pois vamos adicionar
        MOV R9, R4
        MOV R6, R4
        DIV R9, R7
```

```
        MOD R6, R7
        ADD R10, 1
        ADD R9, R8
        ADD R6, R8
        MOVB [R10], R9
        ADD R10, 1
        MOVB [R10], R6
        POP R10
        POP R9
        POP R8
        POP R7
        POP R6
        POP R4
        POP R3
        POP R1
        RET


;Calcula as moedas a retirar para dar o troco
TrocoDoStock:
        PUSH R5
        MOV R10, 0
        MOV R1, 0                       ; R1 tem o valor necessário a incrementar para obter o endereço dos euros ou centimos
        MOV R0, 0                       ; R0 tem o valor necessário a incrementar para obter o endereço dos euros ou centimos
RetirarTrocoDoStock:
        MOVB R6, [R8]                   ; Valor dos euros e posteriormente centimos do troco
CicloRetirarTrocoDoStock:
        MOV R9, 30H
        CMP R6, R9                      ; Enquanto não for 0 ele remove sempre as moedas ou notas mais altas
        JEQ FimCiclo
```

```
        MOV R9, 35H
        MOV R4, 5
        MOV R5, 24A0H
        ADD R5, R1
        CMP R6, R9
        JGE Retirar
        MOV R9, 32H
        MOV R4, 2
        MOV R5, 24B0H
        ADD R5, R0
        CMP R6, R9
        JGE Retirar
        MOV R9, 31H
        MOV R4, 1
        MOV R5, 24C0H
        ADD R5, R0
        CMP R6, R9
        JGE Retirar

Retirar:
        SUB R6, R4          ; Remove o valor que vai ser retirado pela moeda ou nota, por exemplo no troco de 3 euros ele removia uma moeda de 2 primeiro e depois uma de 1
        CALL RemoverDoStock
        JMP CicloRetirarTrocoDoStock
FimCiclo:
        ADD R10, 1          ; Incrementa para saber se já passou a verificação nos centimos e nos euros
        ADD R8, 2           ; Incrementa para ir para os centimos
        MOV R1, 48          ; Incrementar posteriormente o valor de R5 para o endereço dos centimos dessa moeda
        MOV R0, 96          ; Incrementar posteriormente o valor de R5 para o endereço dos centimos dessa moeda
```

```
        CMP R10, 2                          ; Verifica se ja retiramos os
centimos
        JLT RetirarTrocoDoStock             ; Se nao retirou vai retirar
        POP R5
        JMP CicloTalao                      ; Vai para o CicloTalao para mostrar o
mesmo
```