



FCEE - LEI

ARQUITETURA DE COMPUTADORES

Sofia Inácio | Dionísio Barros | Pedro Camacho | Dino Vasconcelos
2023/2024

TRABALHO PRÁTICO 2



Trabalho realizado por:

Ana Leonor Freitas - nº2081821

Diogo Miguel Paixão - nº2079921

19 de abril de 2024

Índice

Introdução.....	3
Objetivos.....	3
Desenvolvimento.....	3
Periféricos de Entrada.....	3
PER_EN.....	3
NPEPE.....	4
Palavra-Passe do Stock.....	4
Periféricos de Saída.....	4
Display.....	4
Funcionamento Geral do Programa.....	4
Menu Inicial.....	4
Menu Comprar.....	4
Menu Confirmação Escolha.....	4
Menu Mostra Preço Total.....	5
Menus Inserir Dinheiro.....	5
Menu Inseriu Dinheiro.....	5
Menu Talão.....	5
Menu Erro Sem Espaço.....	5
Menu Primeira Compra.....	5
Menu Erro Sem Troco.....	5
Menus Devolver Troco.....	5
Menu Agradecimento.....	6
Menu Usar Cartão.....	6
Menu PEPE Não Encontrado.....	6
Menu Consultar Cartão.....	6
Menu Comprar PEPE.....	6
Menu Saldo Insuficiente.....	6
Menu Inserir Saldo PEPE.....	6
Menu Stock.....	7
Menu Palavra-Passe Incorreta.....	7
Menus Consultar Stock.....	7
Funções Extra.....	7
Discussão de resultados.....	7
Conclusão.....	8
Bibliografia.....	8
Anexo A - Fluxogramas.....	9
Anexo B - Funções Extra.....	29
Compra Processada Automaticamente.....	29

Valores Monetários Diferentes.....	30
Botão ON/OFF.....	30
Promoção Primeira Viagem.....	31
Valores monetários de 10 e 20 euros.....	31
Password para acesso ao stock.....	33
Anexo C - Código.....	35
Anexo D - Figuras dos Menus.....	135

Introdução

No segundo trabalho prático de Arquitetura de computadores, foi-nos proposto o desenvolvimento e implementação de um programa em linguagem Assembly para o processador PEPE, com o intuito de criar e gerir bilhetes eletrónicos, denominados PEPE's, para utilização no sistema de viagem de metro.

Será apresentado o desenvolvimento do projeto, juntamente com discussões de resultados e fluxogramas. Para testar o programa, foi utilizado um simulador em JAVA.

Por fim, este projeto visa aprimorar os conteúdos trabalhados nas aulas práticas acerca da linguagem Assembly e o processador PEPE.

Objetivos

O programa desenvolvido tem como objetivo criar um programa em linguagem Assembly para oferecer uma solução eficiente e intuitiva para a geração, recarga e utilização de bilhetes eletrónicos PEPE, proporcionando uma experiência simplificada para os utilizadores do sistema de viagem de metro. Para alcançar esse objetivo, o projeto está dividido em diferentes rotinas, cada uma responsável por uma parte específica do processo, desde:

- Geração de Bilhetes PEPE Únicos
- Gestão de Compras e Recargas
- Displays Intuitiva
- Aceitação de Diferentes Valores Monetários
- Gestão de Stock Monetário:
- Feedback ao Utilizador
- Testes e discussão de resultados
- Fluxogramas

Desenvolvimento

Nesta seção do relatório, iremos descrever os periféricos de entrada e o funcionamento geral do programa em questão.

Periféricos de Entrada

PER_EN

Este é o nosso principal periférico de entrada. Utilizamo-lo para ligar a máquina e seleccionar todas as opções descritas nos menus do programa.

NPEPE

Este é o nosso periférico de inserção do número do PEPE a ser utilizado. Utilizamos-lo no menu de verificação do PEPE para inserir o número a verificar e, após a verificação, entrar no menu de gestão correspondente.

Palavra-Passe do Stock

Este é o nosso periférico de inserção da palavra-passe do stock, utilizado no menu de verificação da palavra-passe do stock. Recebemos a palavra escrita e comparamos com a palavra-passe guardada na memória.

Periféricos de Saída

Display

O nosso único periférico de saída é o display começado no endereço *200H* e acabado no *26FH* com uma dimensão de 7 linhas de 16 caracteres.

Funcionamento Geral do Programa

A máquina de vendas começa desligada. Após ser ligada coloca todas as quantidades dos valores monetários no stock a 10 de forma a existir algum troco na máquina para começar as compras.

Menu Inicial

É nos apresentado o Menu Inicial do programa (*Anexo D - FIG.1*) onde temos a três escolhas nos propostas para esta máquina, que são a *COMPRAR*, o *USAR CARTÃO* e o *STOCK*.

Menu Comprar

Se for escolhida a opção de *COMPRAR*, são reiniciadas todas as variáveis utilizadas na compra (*PrecoTotal*, *ValorInseridoTotal*, *ValorEmFalta*, *Troco*, e todas as moedas de troco), e depois é nos mostrado o Menu de Compra (*Anexo D - FIG.2*) onde aparecem todos os bilhetes disponíveis a comprar e seus respectivos preços.

Menu Confirmação Escolha

Após a escolha de um bilhete no Menu de Compra vai para o menu de confirmação (*Anexo D - FIG.3*) onde questiona se o utilizador quer ou não comprar mais bilhetes, se seleccionar *SIM* volta ao Menu de Compra e escolhe mais bilhetes e se for *NAO* prossegue com a compra.

Menu Mostra Preço Total

Depois de escolher os bilhetes chega ao Menu do Preço Total (*Anexo D - FIG.4*) e mostra o preço total a pagar por esta compra.

Menus Inserir Dinheiro

Após confirmação de compra vai para o Menus de Inserir Dinheiro (*Anexo D - FIG.5*) que são 3 páginas onde temos todas as escolhas monetárias para inserir na máquina desde 10 cêntimos a 20 euros.

Menu Inseriu Dinheiro

Ao inserir um valor monetário na máquina aparece o Menu Inseriu Dinheiro (*Anexo D - FIG.6*) que mostra o valor total inserido até ao momento na máquina e o valor que falta pagar para acabar a compra. Ao inserir dinheiro suficiente na máquina para a compra este menu não aparece e vai logo para o talão.

Menu Talão

Após inserir dinheiro suficiente para completar a compra, vai para o Menu Talão (*Anexo D - FIG.7*), é neste que é criado o cartão PEPE e exibido o seu número, bem como o total da compra completada, e o troco que sobrou dessa.

Menu Erro Sem Espaço

Este menu serve para alertar que não existe mais espaço na máquina para mais nenhum PEPE, logo cancela a compra e volta para o Menu Inicial.

Menu Primeira Compra

Depois do talão é exibido um menu informativo, Menu Primeira Compra (*Anexo D - FIG.8*), que informa o utilizador que a compra será de graça e o dinheiro gasto nos bilhetes é carregado no cartão PEPE criado.

Menu Erro Sem Troco

Este menu serve para alertar que não existe troco suficiente para dar ao utilizador logo vai continuar sem dar esse troco.

Menus Devolver Troco

Após o talão é mostrado os Menus Devolver Troco (*Anexo D - FIG.9*) que são três páginas onde mostra todas as quantidades de valores monetários devolvidos como troco ao utilizador.

Menu Agradecimento

Acaba a parte da compra com um menu de agradecimento pela compra feita. (*Anexo D - FIG.10*)

Menu Usar Cartão

Se for escolhida a opção *USAR CARTÃO*, mostra o menu de verificação do número do PEPE (*Anexo D - FIG.11*) onde é pedido para inserir o número do PEPE (no periférico de entrada NPEPE) e continuar para essa verificação.

Menu PEPE Não Encontrado

Este menu aparece após a verificação se não existir um PEPE com o número introduzido na base de dados dos PEPEs. (*Anexo D - FIG.12*)

Menu Consultar Cartão

Após verificação e depois de achar o PEPE correspondente ao número na base de dados, mostra o menu de gestão do mesmo (*Anexo D - FIG.13*) onde é possível escolher duas ações: a de COMPRAR utilizando o cartão PEPE e a de RECARREGAR o saldo desse cartão.

Menu Comprar PEPE

Ao escolher a opção COMPRAR vamos para um menu completamente igual ao Menu Comprar referido acima.

Faz a mesma rota de compra a partir daqui até ao final só que sem a parte de devolver troco pois é usado o saldo do PEPE para esta.

Menu Saldo Insuficiente

Este menu é exibido na compra com o PEPE após ser verificado que o PEPE não tem saldo suficiente para essa compra. Após a exibição volta para o menu de gestão do PEPE. (*Anexo D - FIG.14*)

Menu Inserir Saldo PEPE

Ao escolher a opção RECARREGAR no menu de gestão do PEPE vamos para um conjunto de menus completamente iguais aos Menus Inserir Dinheiro referidos acima.

Nestes são escolhidos os valores a inserir na máquina para carregar o cartão PEPE e depois de inserir cada valor vai para outra página onde é mostrado o valor total inserido e adicionado ao PEPE. Repete até o utilizador sair e voltar para o menu de gestão do PEPE.

Menu Stock

Se for escolhida a opção *STOCK*, mostra o menu de verificação da palavra-passe (*Anexo D - FIG.15*) onde é pedido para escrever no ecrã (que neste caso é o nosso periférico de entrada para a palavra-passe) a palavra-passe e continuar para a verificação.

Menu Palavra-Passe Incorreta

Este menu aparece após a verificação se a palavra-passe introduzida pelo o utilizador não for igual à guardada na memória. (*Anexo D - FIG.16*)

Menus Consultar Stock

Após verificar que a palavra-passe introduzida é a correta mostra os Menus Consultar Stock (*Anexo D - FIG.17*) que são três páginas onde mostra todas a quantidades dos valores monetários presentes no stock da máquina.

Funções Extra

Em termos de funções extra no nosso conseguimos concretizá-las quase todas com perfeição exceto a na parte de gestão de stock.

Seguem-se aqui essas funções:

- **Gestão do stock de moedas** a ser devolvidas não foi implementada, a nossa solução para esse problema foi que quando não existem moedas suficientes para troco alertamos o cliente que a compra continuará sem devolver o seu troco;
- Após o utilizador inserir um quantia igual ou superior ao valor da compra esta é **processada automaticamente sem introdução de mais inputs/dinheiro**;
- Todos os bilhetes das nossas estações têm **valores monetários distintos**;
- Implementação de um **botão ON/OFF** simples para ligar e desligar a máquina;
- Ao **comprar uma viagem pela primeira vez**, essa é grátis e o seu valor é adicionado ao saldo do cartão PEPE criado;
- **Valores monetários de 10 e 20 euros** introduzidos no sistema;
- Utilização de uma **password para o acesso ao stock** da máquina.

Discussão de resultados

Após a conclusão e teste do projeto, consideramos que foi um sucesso. Implementamos todas as funcionalidades solicitadas, além de quase todas as funcionalidades extras, com exceção da gestão do stock.

Em termos de desempenho, o projeto opera corretamente do início ao fim, sem a ocorrência de erros não detetados e tratados.

Conclusão

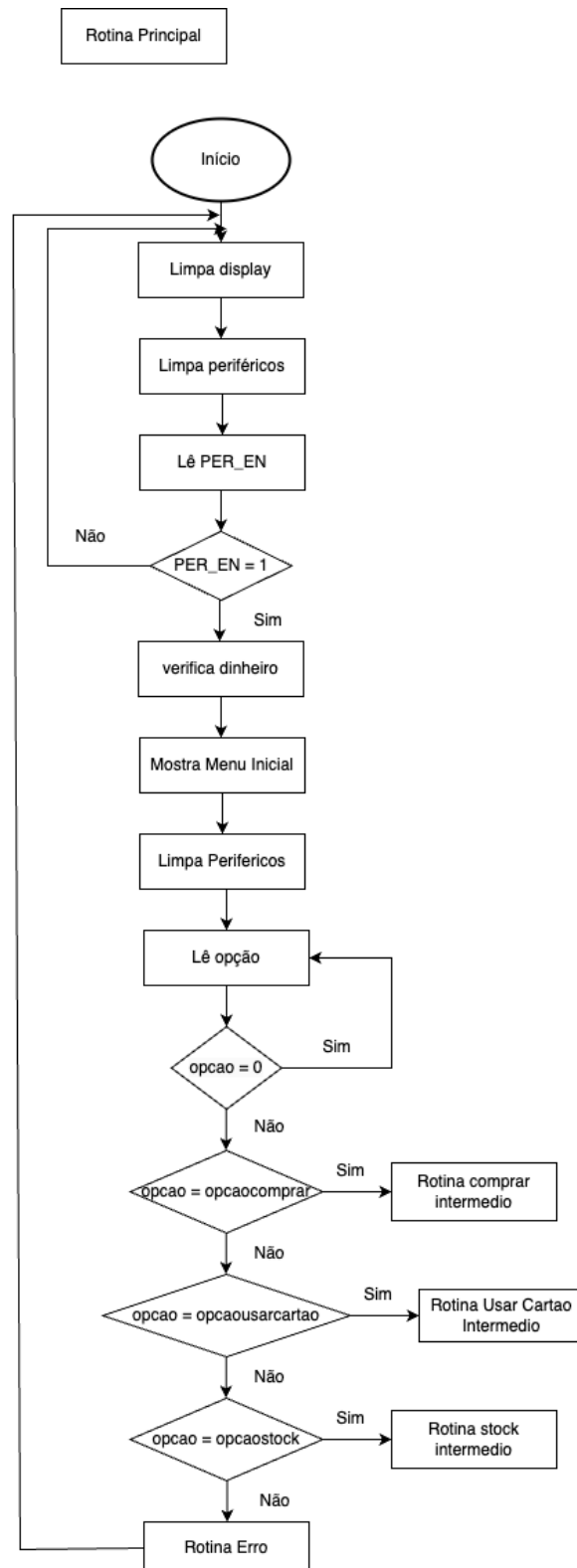
Concluindo, o projeto alcançou seus objetivos de forma satisfatória. O código foi desenvolvido de maneira clara e livre de erros, demonstrando nossa compreensão e aplicação dos conhecimentos adquiridos, especialmente na linguagem Assembly.

Esta experiência foi valiosa para consolidar nosso entendimento teórico e desenvolver habilidades práticas relevantes, preparando-nos para desafios futuros na área de arquitetura de computadores.

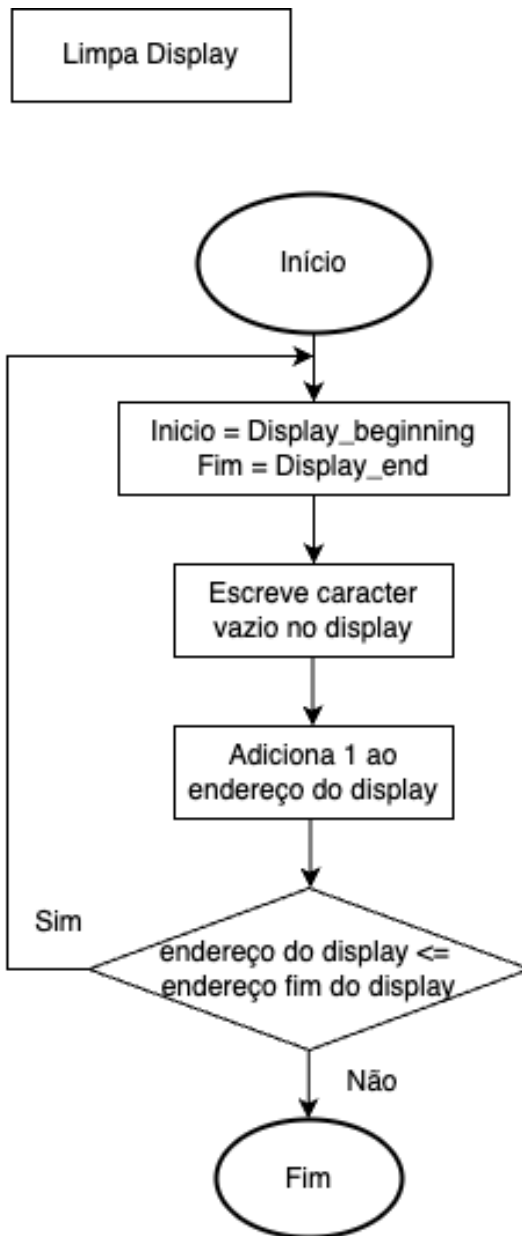
Bibliografia

Materiais de referência utilizados incluem as práticas laboratoriais, seus exercícios, e slides fornecidos durante o seu curso.

Anexo A - Fluxogramas

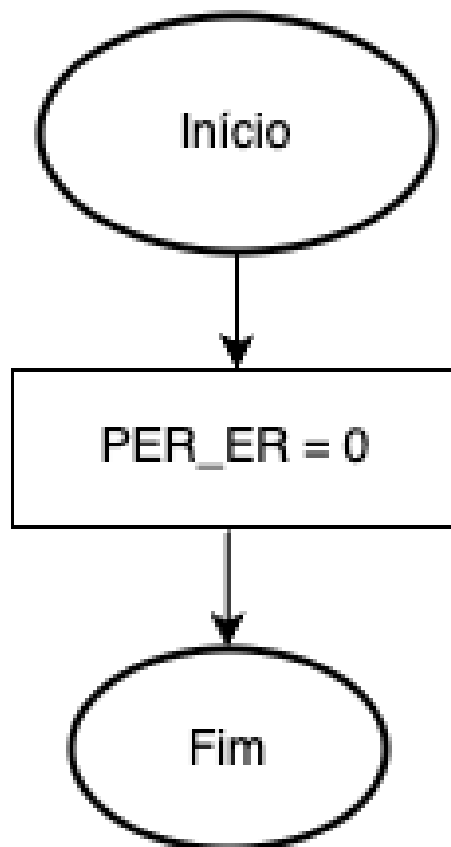


Fluxograma da Rotina Principal



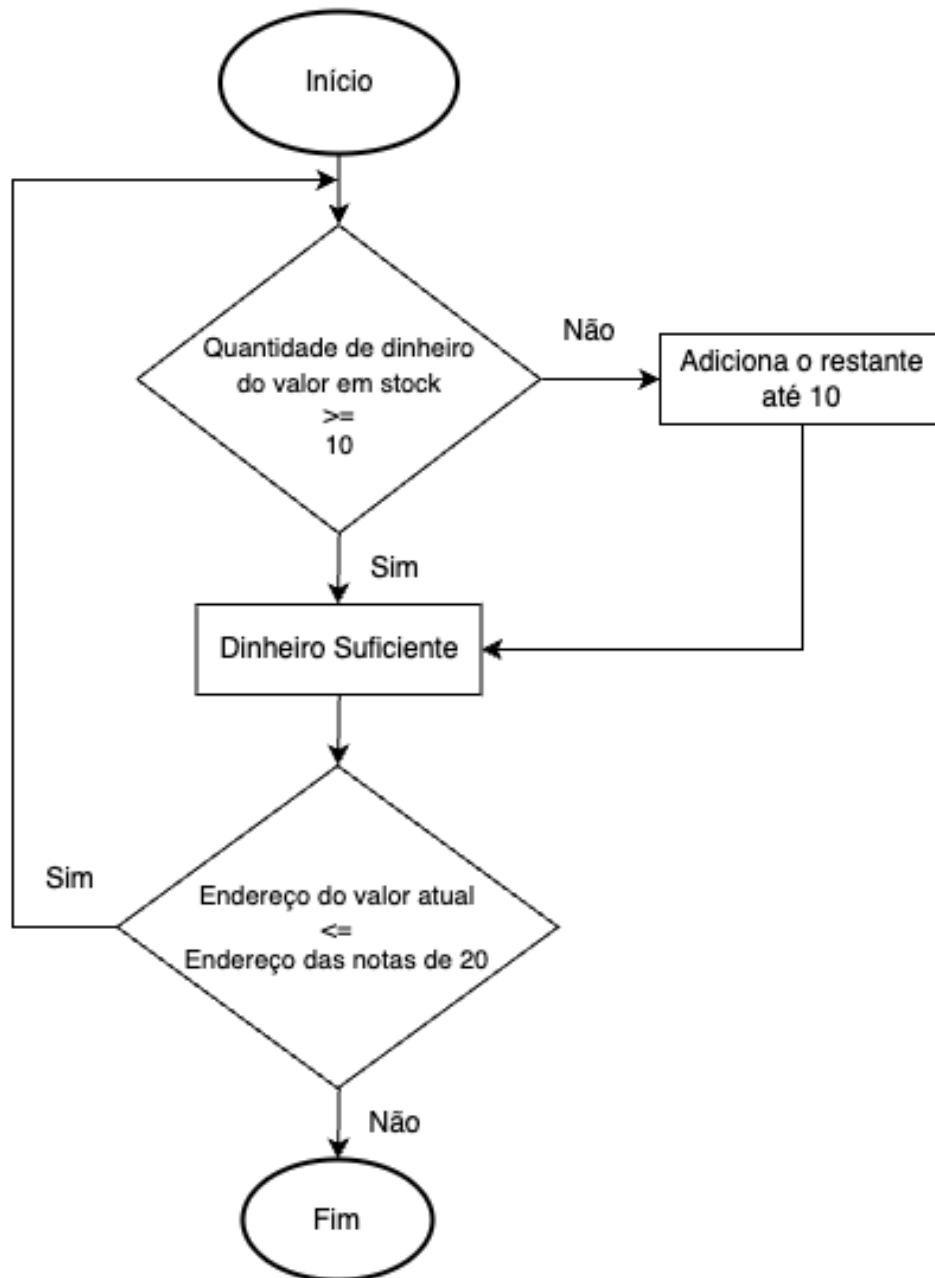
Fluxograma da Rotina Limpa Display

Limpa perifericos

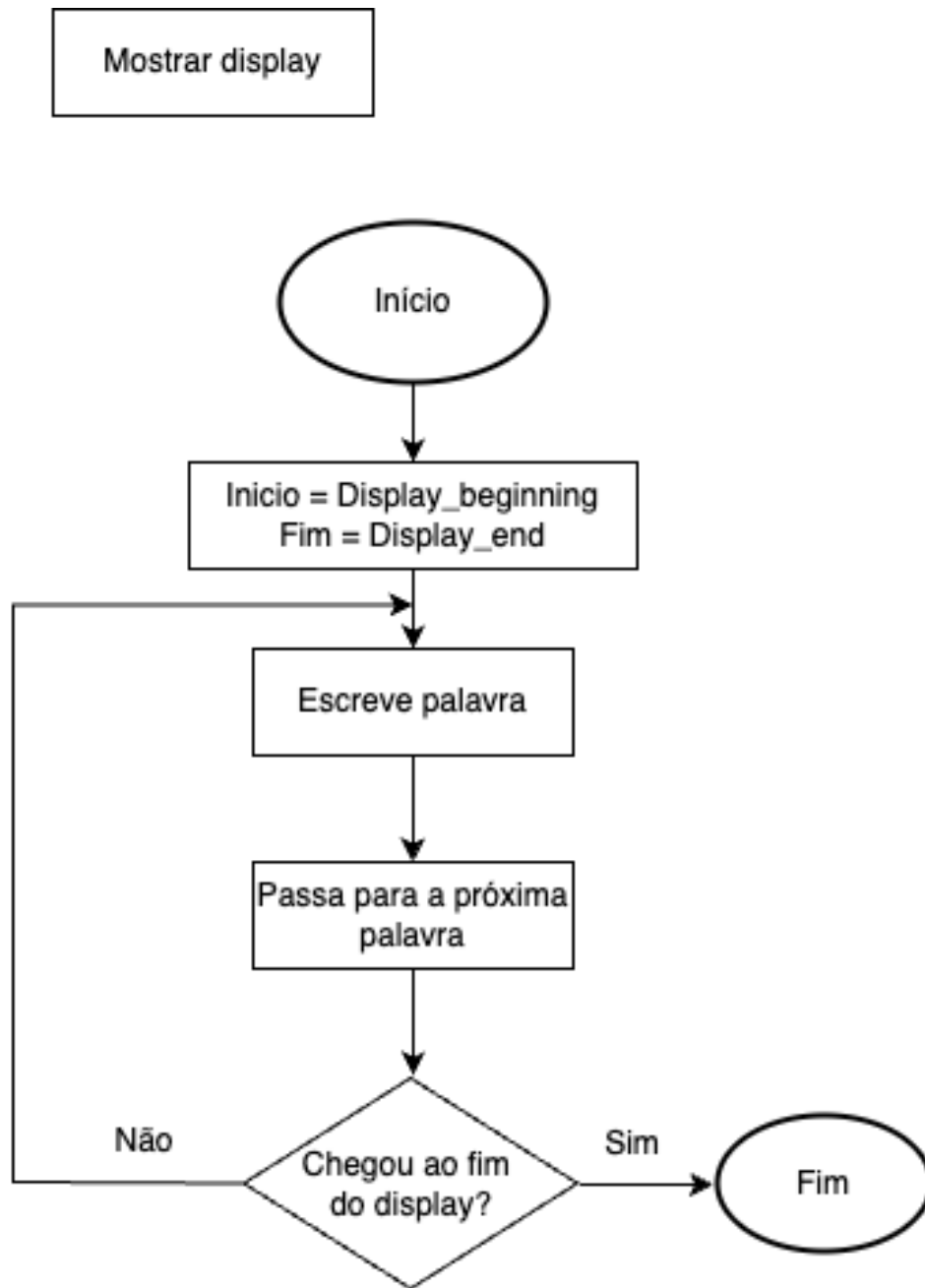


Fluxograma da Rotina Limpa Periféricos

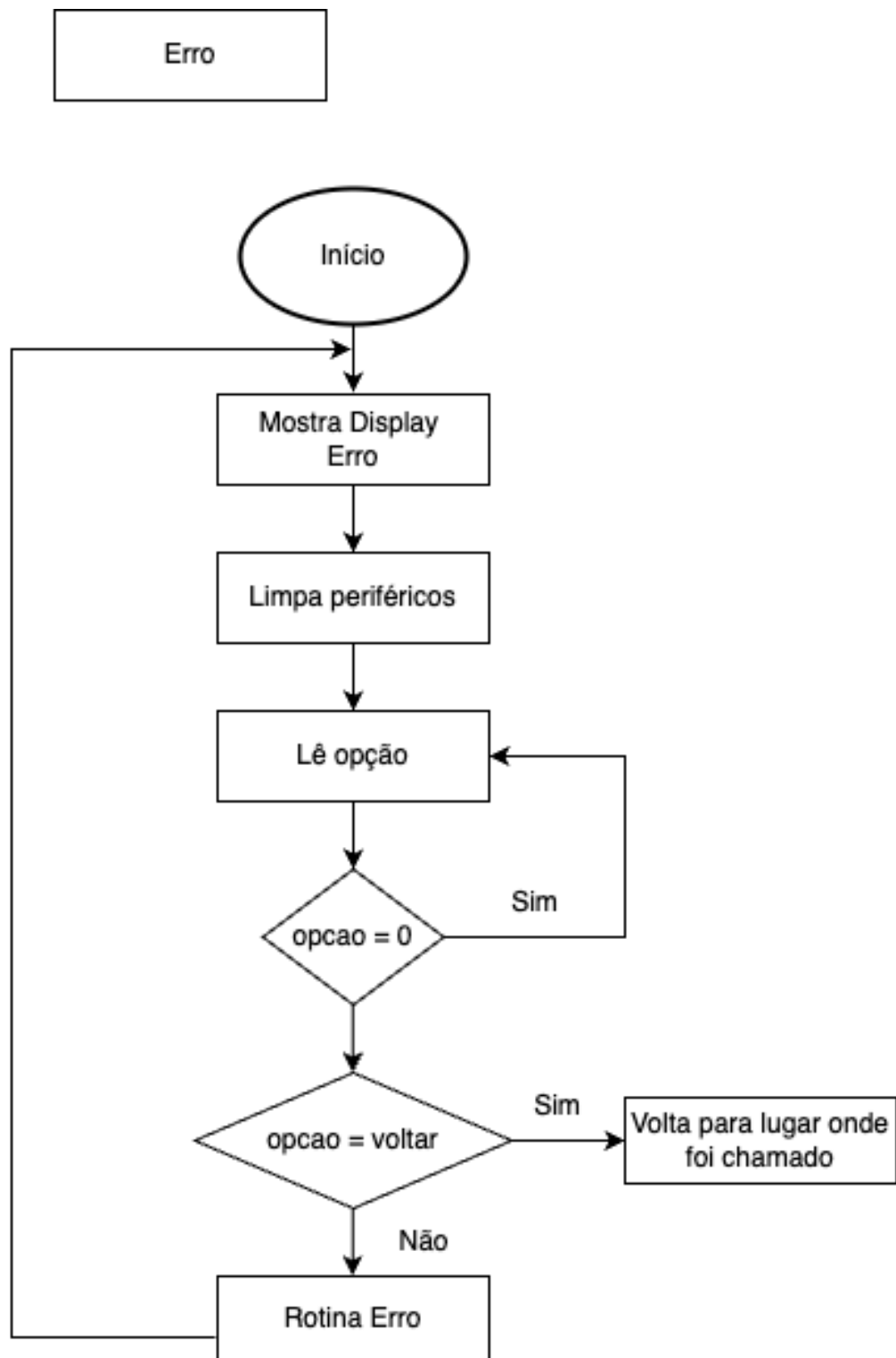
Verifica Dinheiro



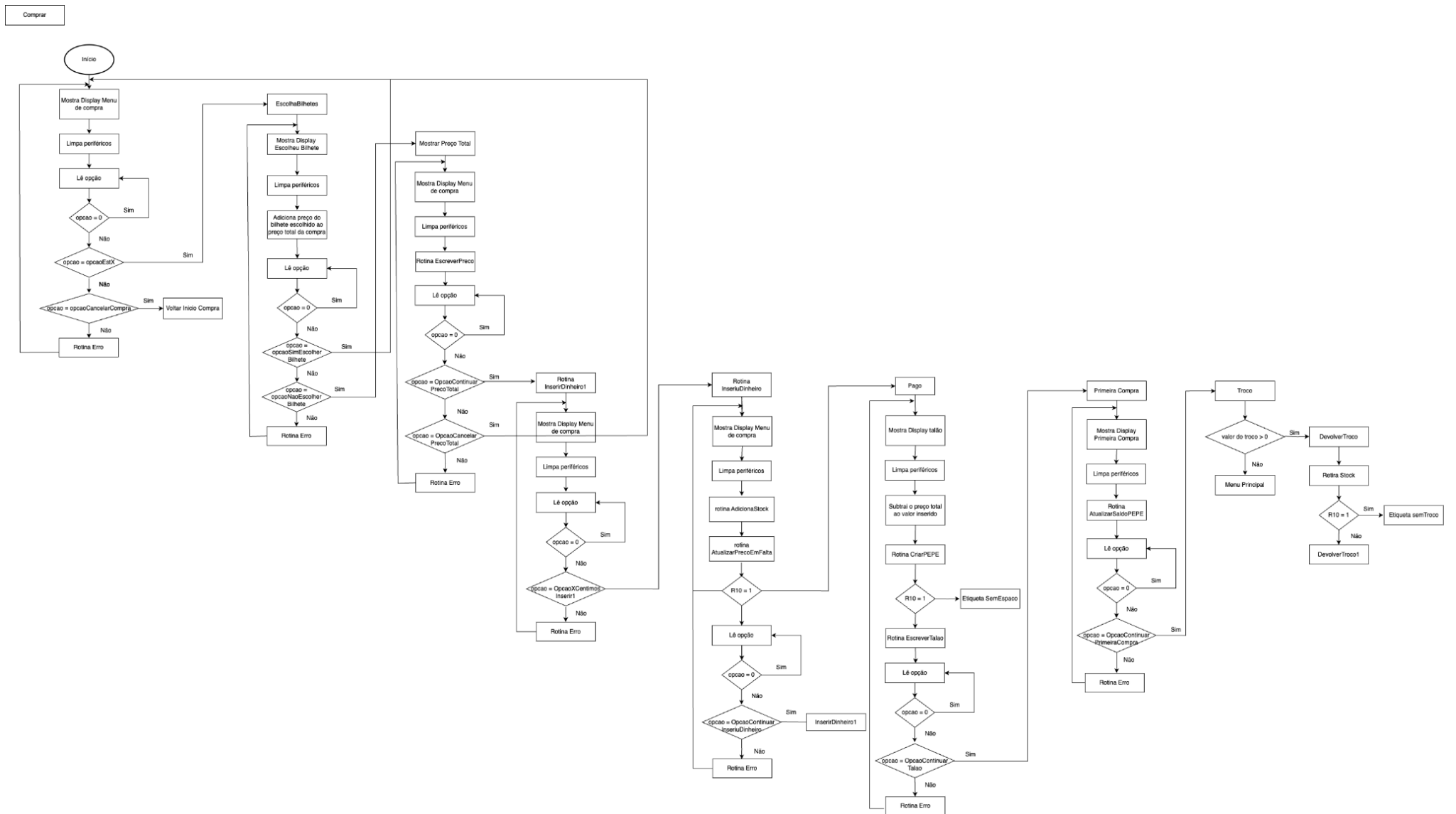
Fluxograma da Rotina Verifica Dinheiro



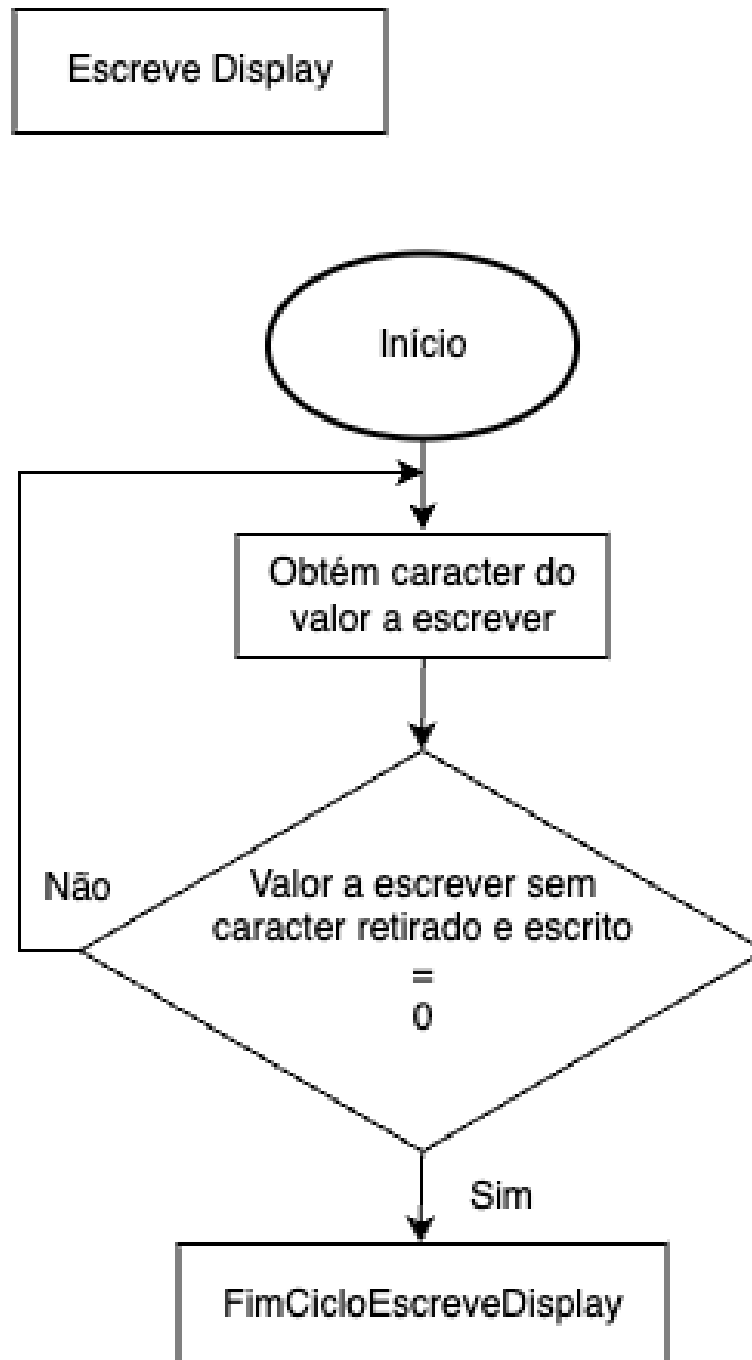
Fluxograma da Rotina Mostra Display



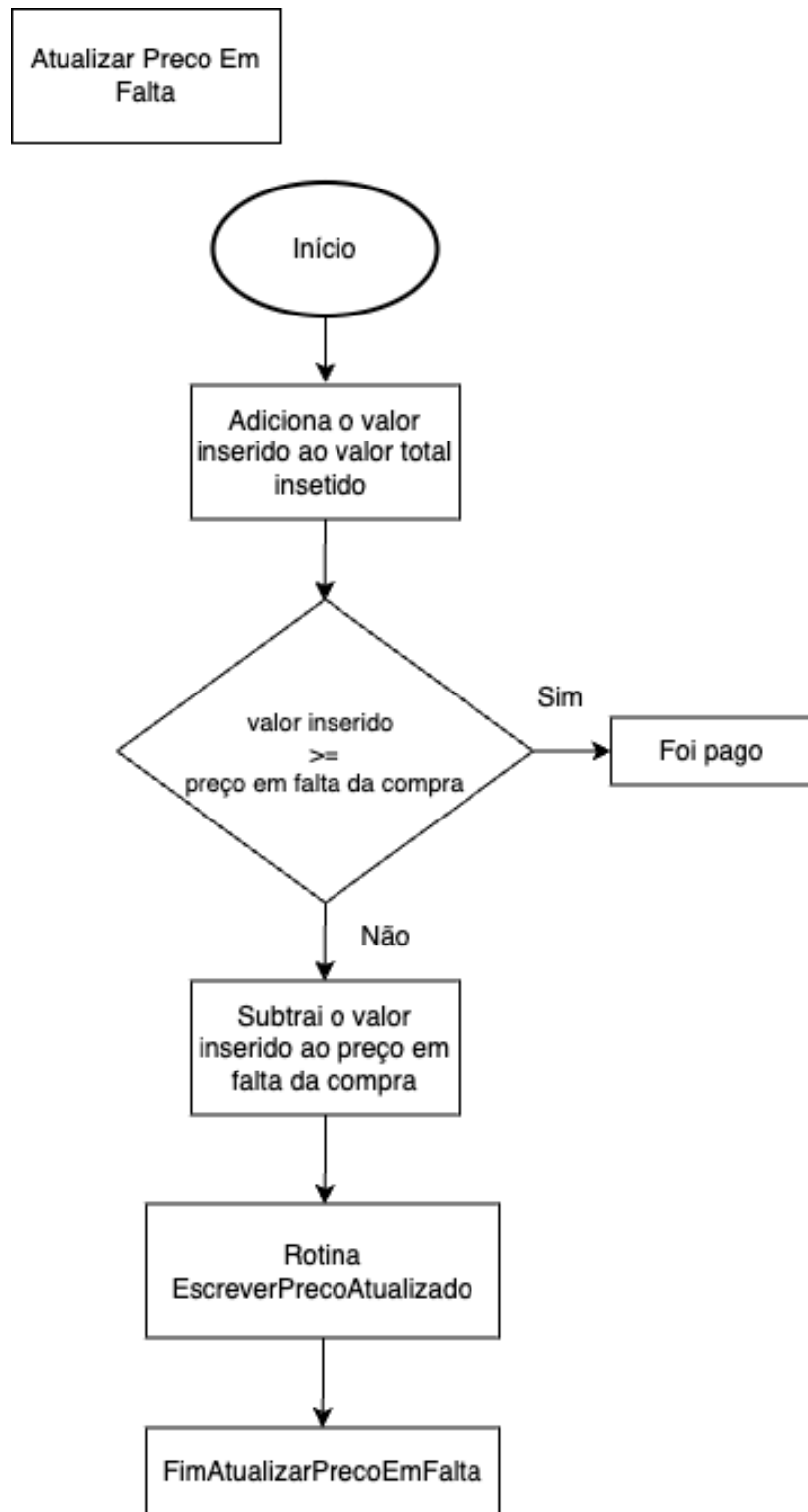
Fluxograma da Rotina Erro



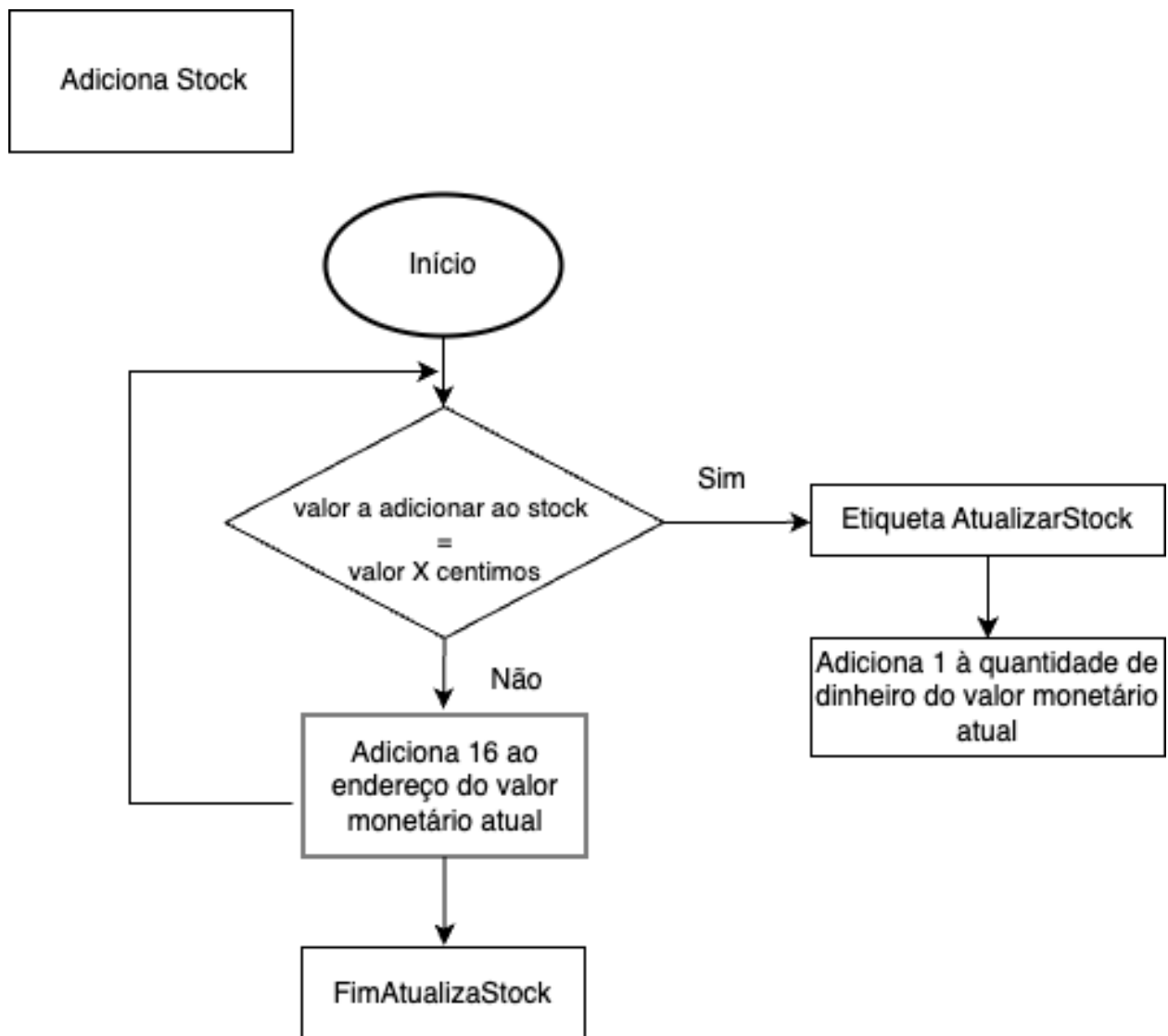
Fluxograma da Rotina Comprar



Fluxograma da Rotina Escreve Display

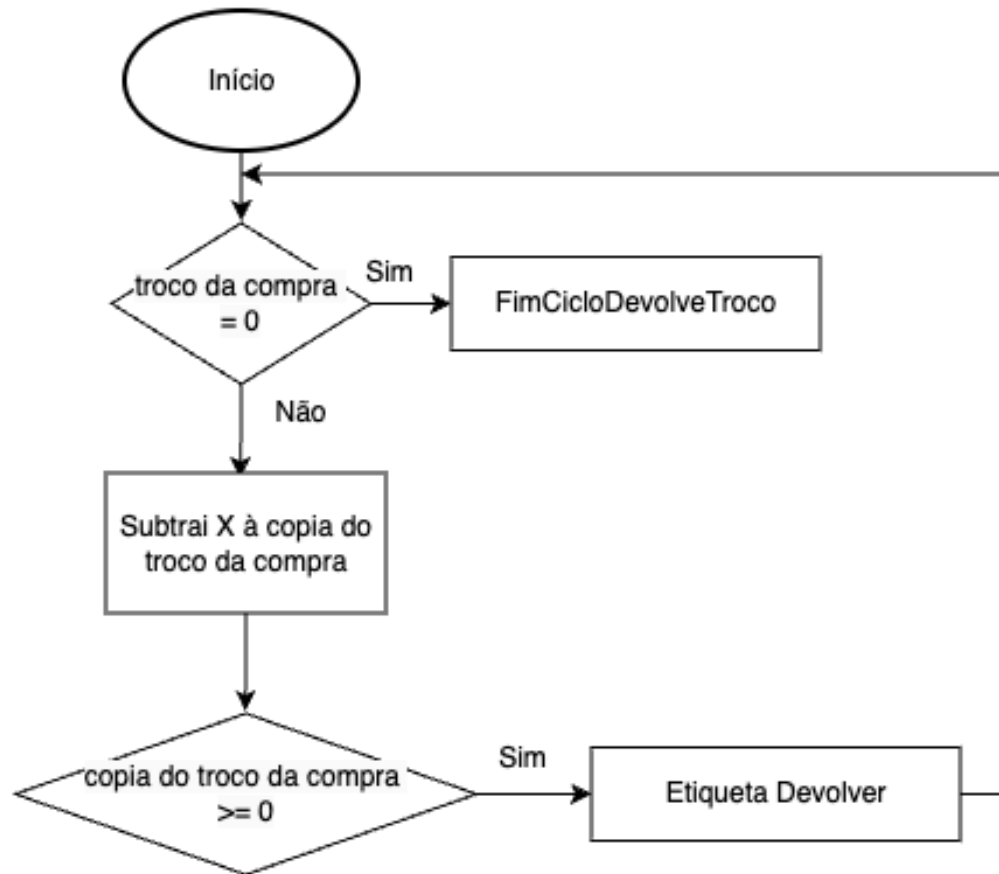


Fluxograma da Rotina Atualizar Preço em Falta

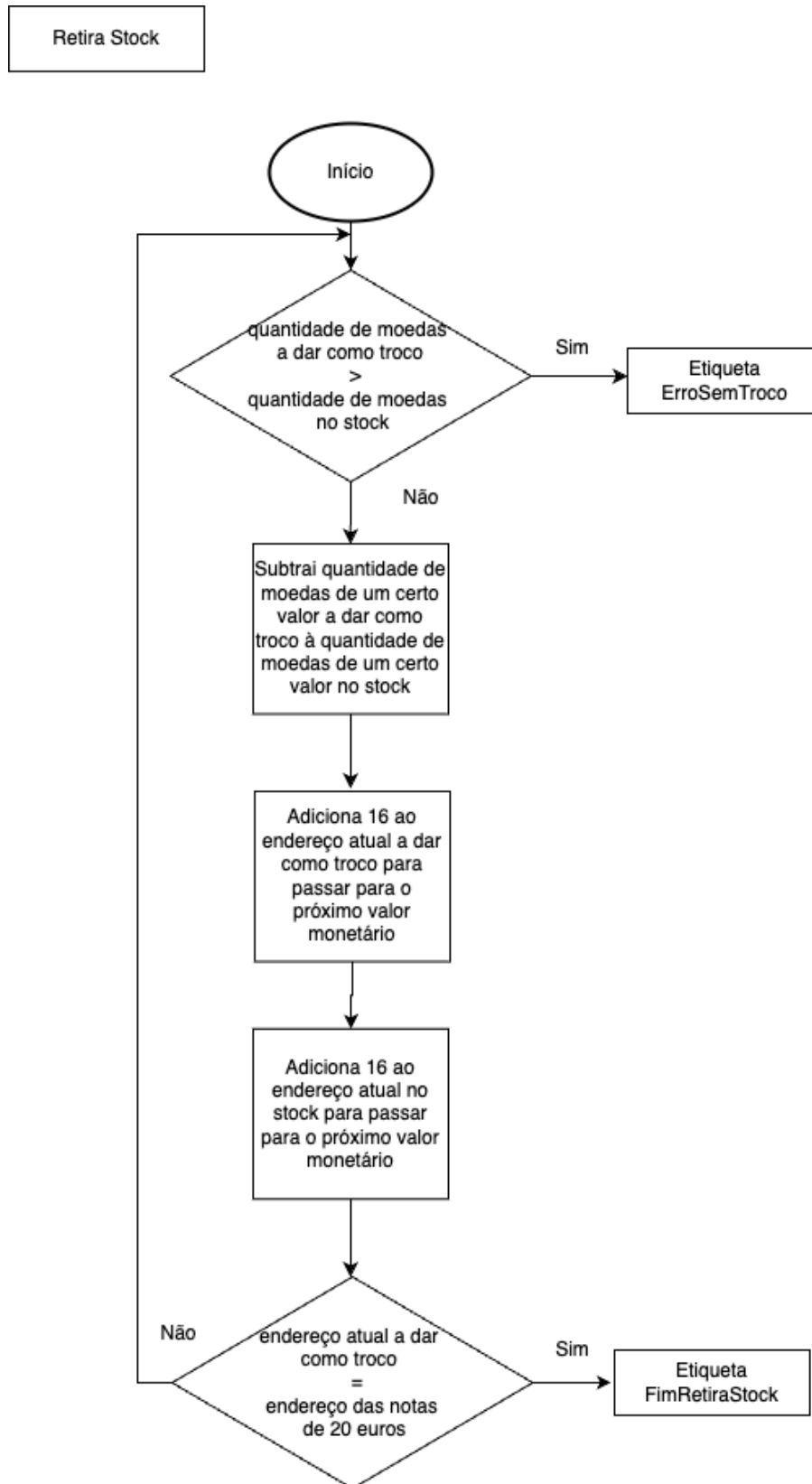


Fluxograma da Rotina Adiciona Stock

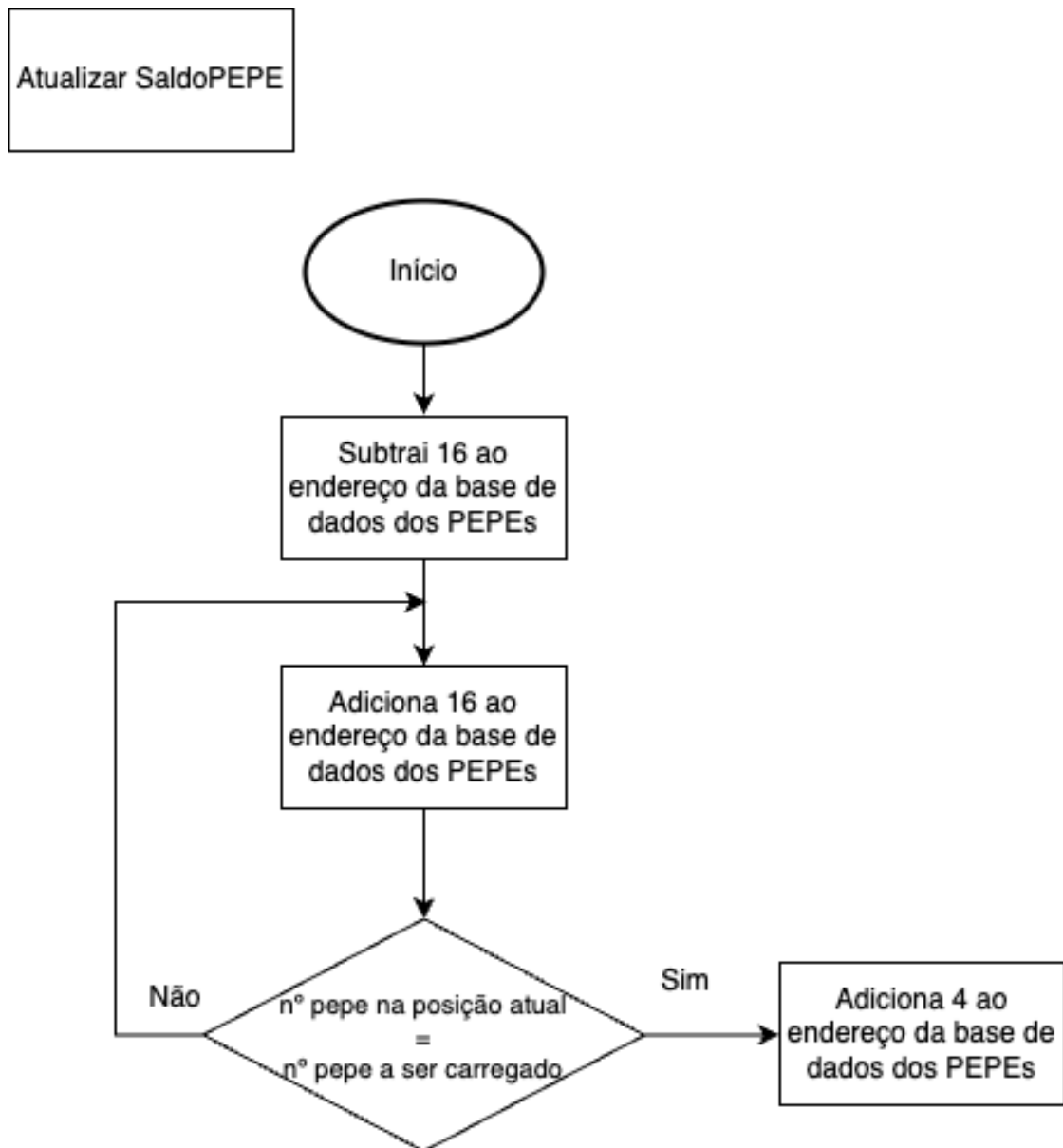
Devolve Troco



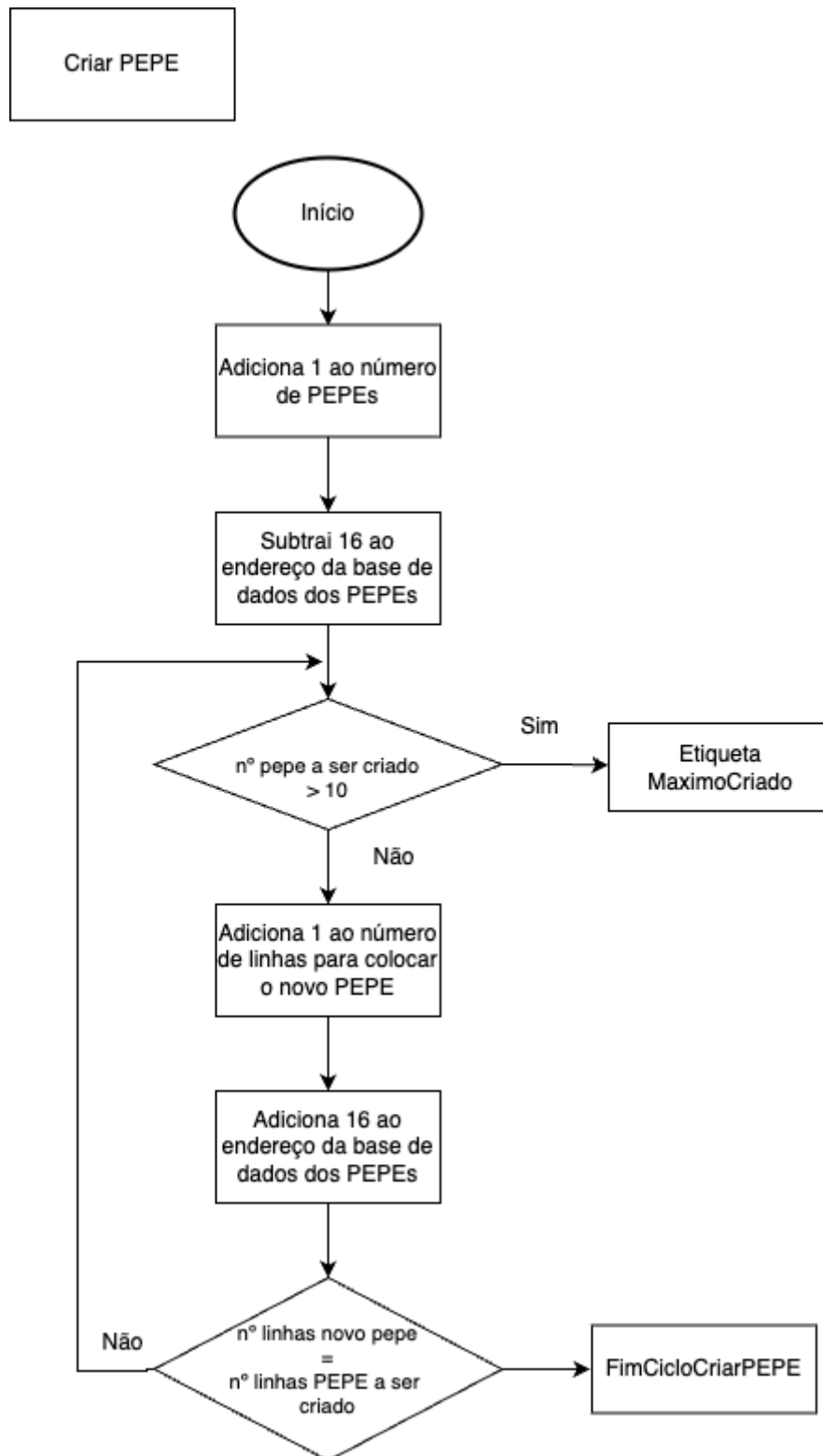
Fluxograma da Rotina Devolve Troco



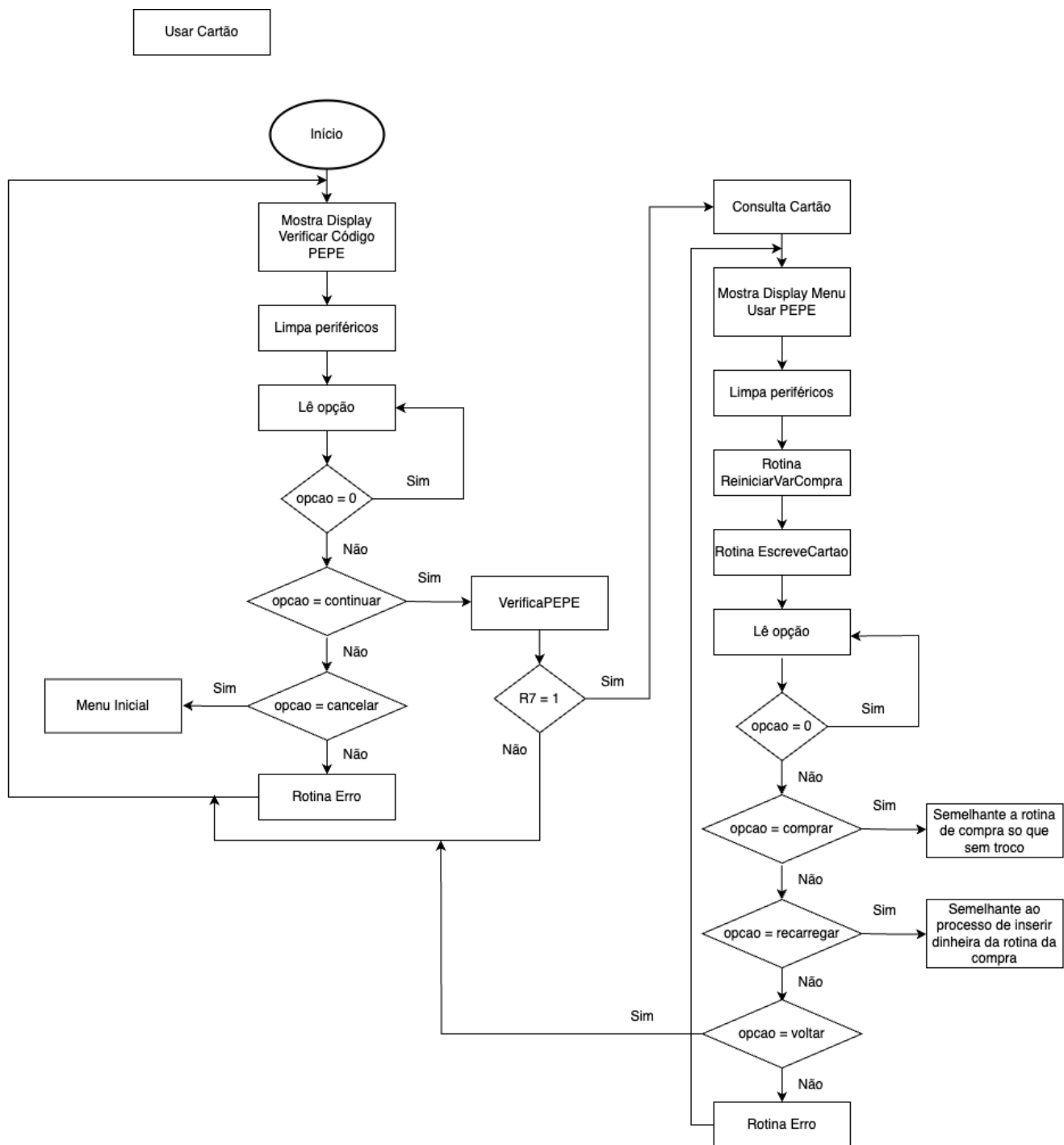
Fluxograma da Rotina Retira Stock



Fluxograma da Rotina Atualizar Saldo PEPE

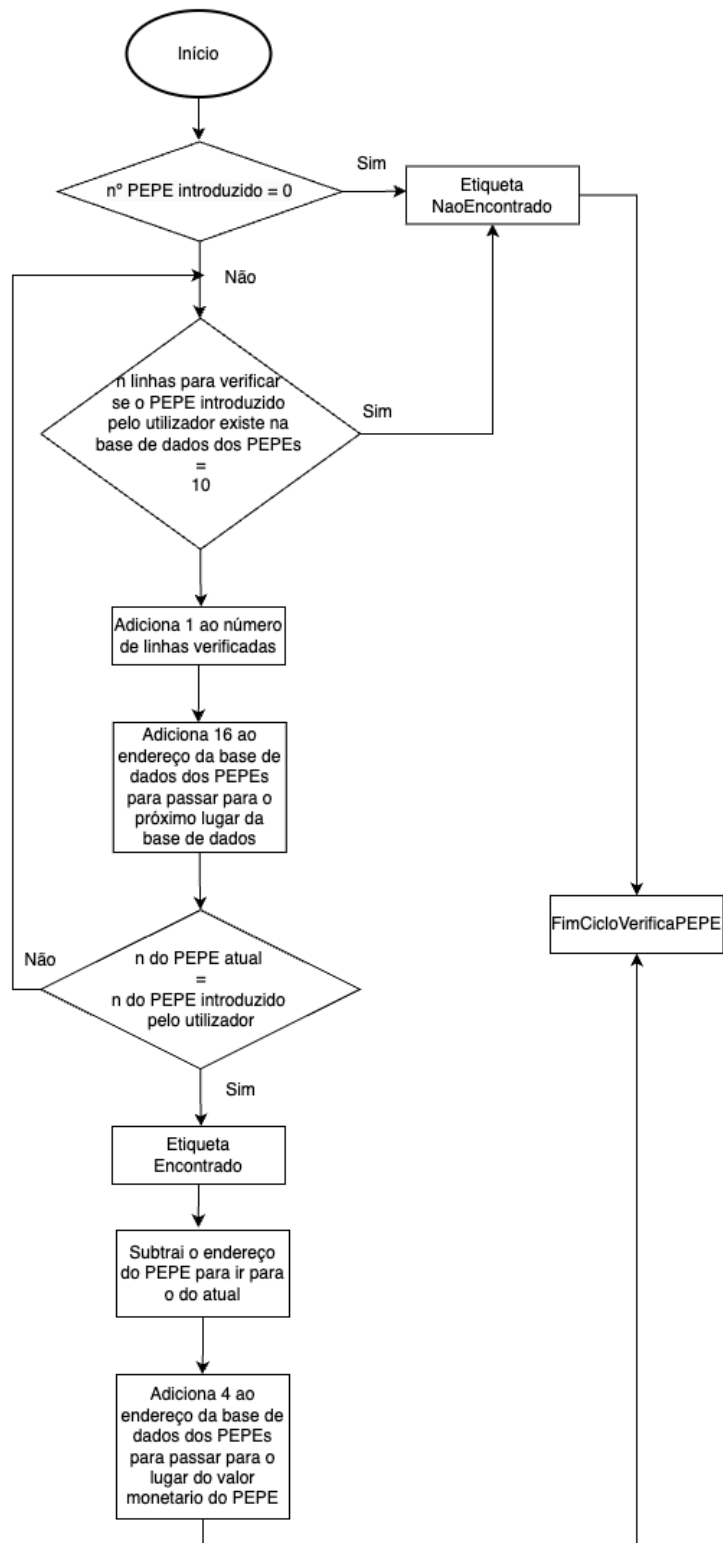


Fluxograma da Rotina Criar PEPE

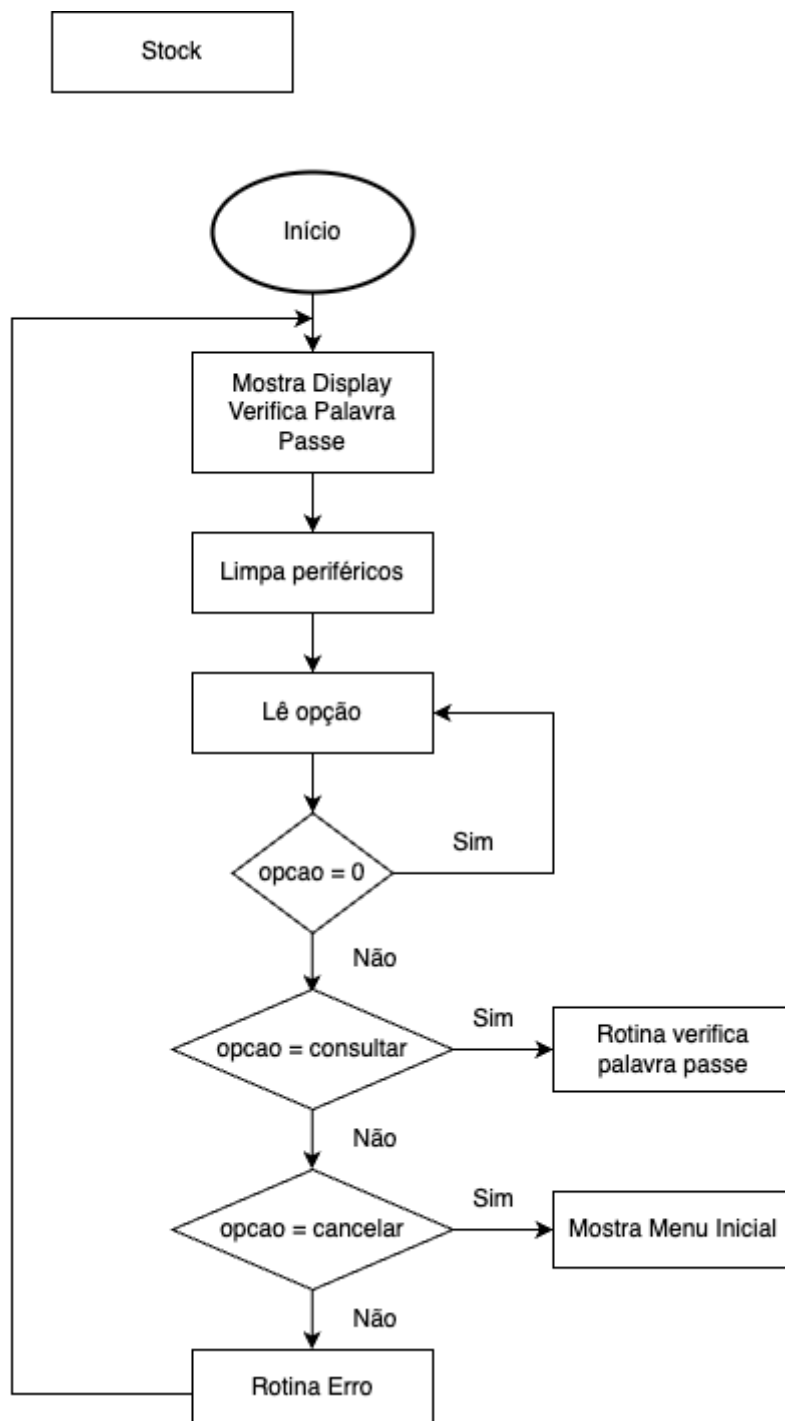


Fluxograma da Rotina Usar Cartão

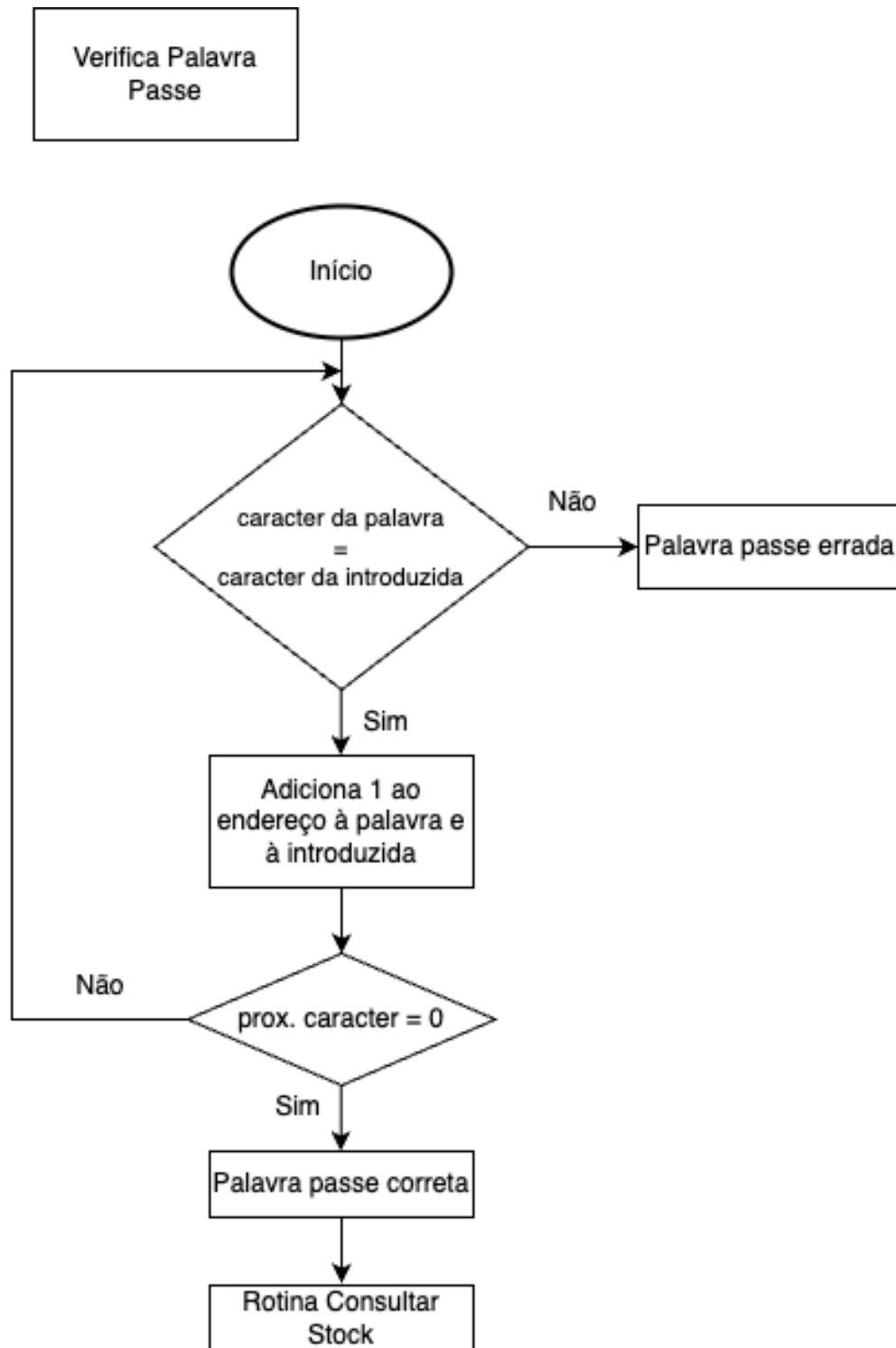
Verifica PEPE



Fluxograma da Rotina Verificar PEPE

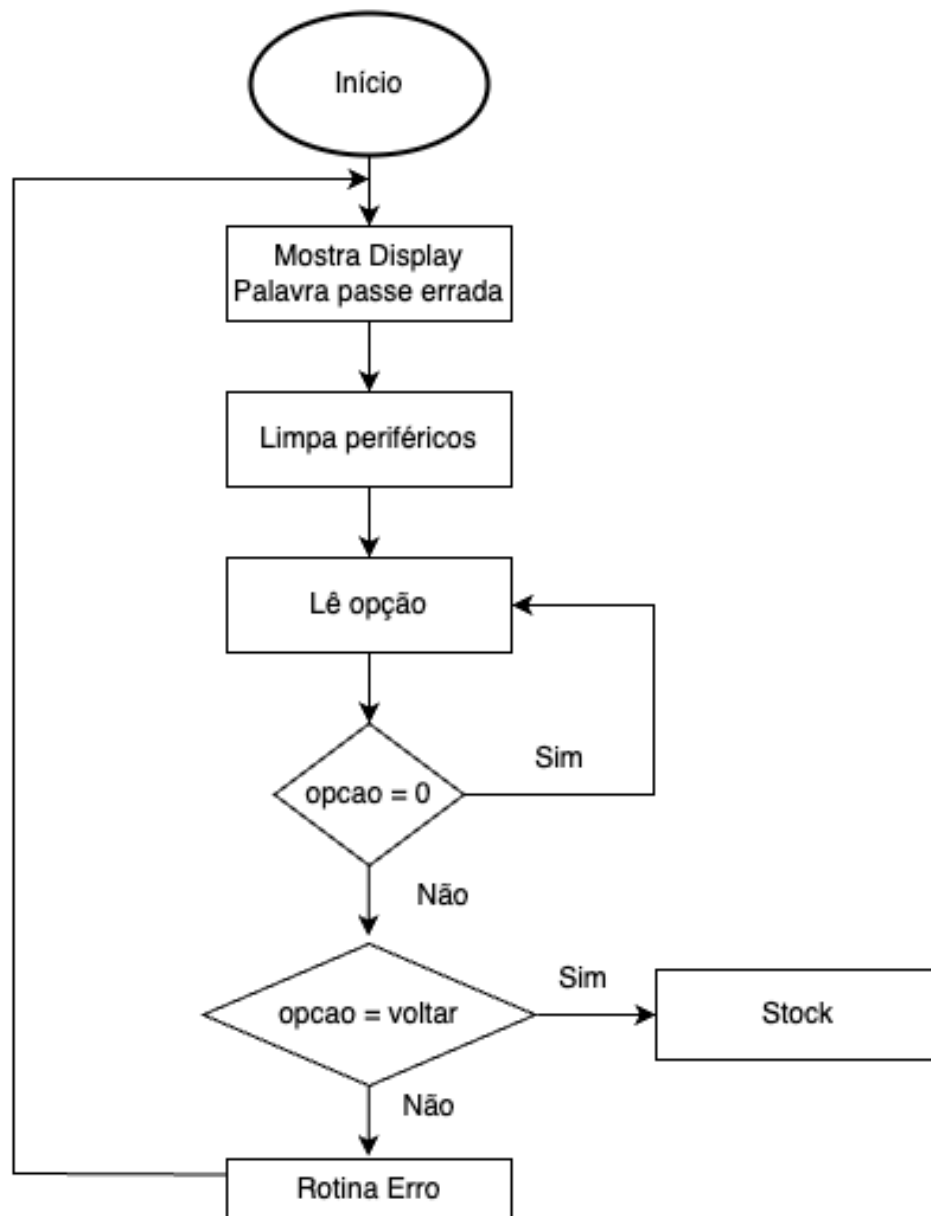


Fluxograma da Rotina Stock

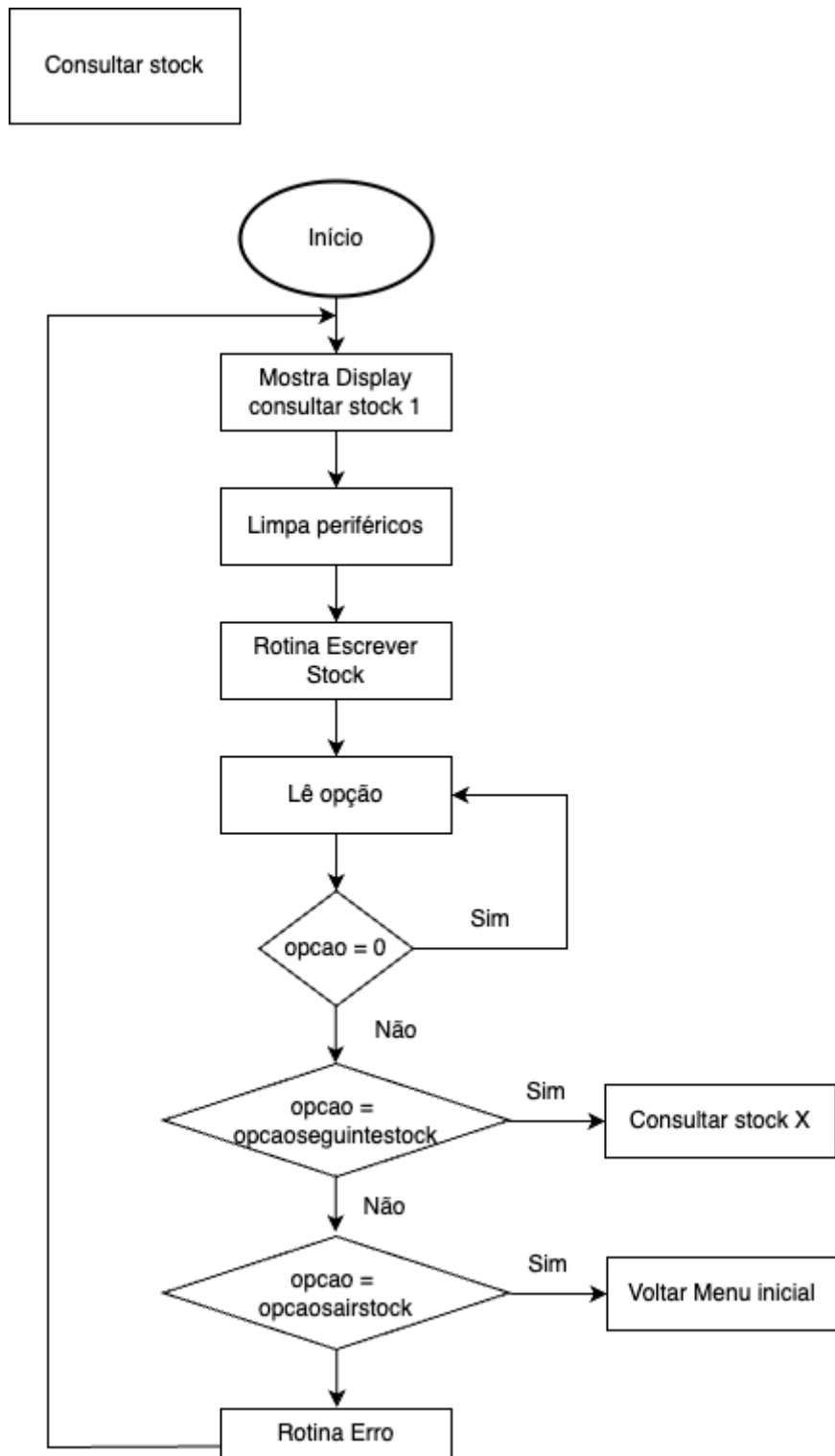


Fluxograma da Rotina Verifica Palavra Passe

Palavra Passe Errada



Fluxograma da Rotina Palavra Passe Errada



Fluxograma da Rotina Consultar stock

Anexo B - Funções Extra

Compra Processada Automaticamente

InseriuDinheiro:

```
    MOV R2, Display_InseriuDinheiro          ; R2 tem o endereço do display
da inserção de dinheiro
    CALL MostraDisplay                        ; Mostra o display da inserção
de dinheiro
    CALL LimpaPerifericos                    ; Limpa os periféricos

    CALL AdicionaStock                       ; Chama a rotina AdicionaStock,
que adiciona o valor do dinheiro inserido ao stock da máquina

    MOV R10, 0                              ; R10 tem o valor 0 para
sinalizar que o preço em falta não foi pago

    CALL AtualizarPrecoEmFalta               ; Chama a rotina
AtualizarPrecoEmFalta, que atualiza o preço em falta da compra e o valor total
inserido pelo utilizador

    CMP R10, 1                              ; Compara o valor de R10 com 1,
que quer dizer que o preço em falta foi pago
    JEQ Pago                                ; Se o valor de R10 for 1, salta
para a etiqueta Pago
```

LeOpcaoInseriuDinheiro:

```
    MOV R0, PER_EN                          ; R0 tem o endereço do
periférico de entrada
    MOVB R1, [R0]                           ; R1 tem o valor do periférico
de entrada
    CMP R1, 0                               ; Compara o valor do periférico
de entrada com 0
    JEQ LeOpcaoInseriuDinheiro              ; Se o valor do periférico de
entrada for 0, repete o ciclo

    CMP R1, OpcaoContinuarInseriuDinheiro   ; Compara a opção escolhida pelo
utilizador com a opção de continuar
    JEQ InserirDinheiro1                     ; Se a opção escolhida pelo
utilizador for a opção de continuar, salta para a etiqueta InserirDinheiro1

    CALL RotinaErro                          ; Se a opção escolhida pelo
utilizador não for a opção de continuar, chama a rotina RotinaErro
```

```

    JMP InserirDinheiro          ; Salta para a etiqueta
InserirDinheiro novamente

```

Valores Monetários Diferentes

```

;Preços dos
bilhetes-----
-----

BEstacao1Preco EQU 150          ; Preço do bilhete da estação 1
BEstacao2Preco EQU 250          ; Preço do bilhete da estação 2
BEstacao3Preco EQU 400          ; Preço do bilhete da estação 3
BEstacao4Preco EQU 550          ; Preço do bilhete da estação 4
BEstacao5Preco EQU 700          ; Preço do bilhete da estação 5

```

Botão ON/OFF

Princípio:

```

    MOV SP, StackPointer        ; Inicializa o Stack Pointer
    CALL LimpaDisplay           ; Limpa o display
    CALL LimpaPerifericos       ; Limpa os periféricos

```

LePER:

```

    MOV R0, PER_EN              ; R0 tem o endereço do botão
PER_EN
    MOVB R1, [R0]                ; Leitura do periférico PER_EN
    CMP R1, 1                    ; Compara o valor do botão PER_EN
com 1
    JNE LePER                    ; Se o valor do botão PER_EN for
diferente de 1, repete a leitura
    CALL VerificaDinheiro        ; Verifica a quantidade de cada
tipo monetário no stock de modo a ter algum suficiente no início da máquina

```

Ligado:

```

    MOV R2, Display_MenuInicial ; R2 tem o endereço do menu
inicial
    CALL MostraDisplay           ; Mostra o menu inicial no
display
    CALL LimpaPerifericos       ; Limpa os periféricos

```

Promoção Primeira Viagem

```
;Promoção da Primeira
Compra-----
-----
PrimeiraCompra:

    MOV R2, Display_PrimeiraCompra      ; R2 tem o endereço do display
da primeira compra
    CALL MostraDisplay                   ; Mostra o display da primeira
compra
    CALL LimpaPerifericos                ; Limpa os periféricos

    MOV R5, PrecoTotalCompra             ; R5 tem o endereço do preço
total da compra
    MOV R10, [R5]                       ; R10 tem o valor do preço total
da compra
    CALL AtualizarSaldoPEPE              ; Chama a rotina
AtualizarSaldoPEPE, que atualiza o saldo do PEPE atual

LeOpcaoPrimeiraCompra:

    MOV R0, PER_EN                      ; R0 tem o endereço do
periférico de entrada
    MOVB R1, [R0]                       ; R1 tem o valor do periférico
de entrada
    CMP R1, 0                           ; Compara o valor do periférico
de entrada com 0
    JEQ LeOpcaoPrimeiraCompra           ; Se o valor do periférico de
entrada for 0, repete o ciclo

    CMP R1, OpcaoContinuarPrimeiraCompra ; Compara a opção escolhida pelo
utilizador com a opção de continuar
    JEQ Troco                           ; Se a opção escolhida pelo
utilizador for a opção de continuar, salta para a etiqueta Troco

    CALL RotinaErro                      ; Se a opção escolhida pelo
utilizador não for a opção de continuar, chama a rotina RotinaErro
    JMP PrimeiraCompra                  ; Salta para a etiqueta
PrimeiraCompra novamente
```

Valores monetários de 10 e 20 euros

```
InserirDinheiro3:
```



```

    MOV R2, Display_InserirDinheiro3          ; R2 tem o endereço do display
da inserção de dinheiro
    CALL MostraDisplay                        ; Mostra o display da inserção
de dinheiro
    CALL LimpaPerifericos                    ; Limpa os periféricos

LeOpcaoInserirDinheiro3:

    MOV R0, PER_EN                          ; R0 tem o endereço do
periférico de entrada
    MOVB R1, [R0]                           ; R1 tem o valor do periférico
de entrada
    CMP R1, 0                               ; Compara o valor do periférico
de entrada com 0
    JEQ LeOpcaoInserirDinheiro3              ; Se o valor do periférico de
entrada for 0, repete o ciclo

    MOV R5, Valor10Euros                    ; R5 tem o valor de 10 euros
    CMP R1, Opcao10EuroInserir3             ; Compara a opção escolhida pelo
utilizador com a opção de 10 euros
    JEQ InseriuDinheiro                      ; Se a opção escolhida pelo
utilizador for a opção de 10 euros, salta para a etiqueta InseriuDinheiro

    MOV R5, Valor20Euros                    ; R5 tem o valor de 20 euros
    CMP R1, Opcao20EuroInserir3             ; Compara a opção escolhida pelo
utilizador com a opção de 20 euros
    JEQ InseriuDinheiro                      ; Se a opção escolhida pelo
utilizador for a opção de 20 euros, salta para a etiqueta InseriuDinheiro

    CMP R1, OpcaoSeguinteInserir3           ; Compara a opção escolhida pelo
utilizador com a opção de seguinte
    JEQ InserirDinheiro1                     ; Se a opção escolhida pelo
utilizador for a opção de seguinte, salta para a etiqueta InserirDinheiro1

    CMP R1, OpcaoCancelarInserir3           ; Compara a opção escolhida pelo
utilizador com a opção de cancelar
    JEQ VoltarInicioCompra2                 ; Se a opção escolhida pelo
utilizador for a opção de cancelar, salta para a etiqueta VoltarInicioCompra2

    CALL RotinaErro                          ; Se a opção escolhida pelo
utilizador não for nenhuma das opções anteriores, chama a rotina RotinaErro
    JMP InserirDinheiro3                     ; Salta para a etiqueta
InserirDinheiro3 novamente

```

Password para acesso ao stock

```
;Verificação da Palavra
Passe-----
-----

VerificacaoStock:

    MOV R2, Display_VerificacaoStock          ; R2 tem o endereço do display
para autenticação no stock
    CALL MostraDisplay                        ; Mostra o display de
autenticação no stock
    CALL LimpaPerifericos                    ; Limpa os periféricos

LePalavraPasseStock:

    MOV R0, PER_EN                            ; R0 tem o endereço do botão
opção
    MOVB R1, [R0]                             ; Leitura do periférico Opção
    CMP R1, 0                                 ; Compara o valor do botão opção
com 0
    JEQ LePalavraPasseStock                  ; Se o valor do botão for 0,
refaz a leitura

    CMP R1, OpcaoConsultarVerificarStock      ; Compara o valor do botão opção
com a opção de consultar stock
    JEQ ConsultarVerificarStock               ; Se o valor do botão for igual
à opção de consultar stock, salta para a etiqueta ConsultarVerificarStock

    CMP R1, OpcaoCancelarVerificarStock       ; Compara o valor do botão opção
com a opção de cancelar
    JEQ VoltarInicioStock                    ; Se o valor do botão for igual
à opção de cancelar, salta para o menu principal

    CALL RotinaErro                          ; Se o valor do botão não for
nenhuma das opções anteriores, chama a rotina RotinaErro
    JMP VerificacaoStock                     ; Salta para a etiqueta
VerificacaoStock novamente

ConsultarVerificarStock:

    MOV R6, 0                                ; R6 tem o valor 0
    CALL VerificaPalavraPasseStock            ; Chama a rotina
VerificaPalavraPasseStock, que verifica se a palavra-passe introduzida está correta
    CMP R6, 1                                ; Compara o valor de R6 com 1,
que quer dizer que a palavra-passe está correta
```

```

    JEQ ConsultarStock                ; Se o valor de R6 for 1, salta
para a etiqueta ConsultarStock

;Palavra-Passe
Incorreta-----
-----
PalavraPasseIntroduzidaErrada:

    MOV R2, Display_PalavraPasseErrada    ; R2 tem o endereço do display
de palavra-passe errada
    CALL MostraDisplay                    ; Mostra o display de
palavra-passe errada
    CALL LimpaPerifericos                ; Limpa os periféricos

LeOpcaoPalavraPasseIntroduzidaErrada:

    MOV R0, PER_EN                      ; R0 tem o endereço do botão
opção
    MOVB R1, [R0]                        ; Leitura do periférico Opção
    CMP R1, 0                            ; Compara o valor do botão opção
com 0
    JEQ LeOpcaoPalavraPasseIntroduzidaErrada ; Se o valor do botão for 0,
refaz a leitura

    CMP R1, OpcaoVoltarErro              ; Compara o valor do botão opção
com a opção de voltar
    JEQ VerificacaoStock                  ; Se o valor do botão for igual
à opção de voltar, salta para a etiqueta VerificacaoStock

    CALL RotinaErro                       ; Se o valor do botão não for
nenhuma das opções anteriores, chama a rotina RotinaErro
    JMP PalavraPasseIntroduzidaErrada      ; Salta para a etiqueta
PalavraPasseIntroduzidaErrada novamente

```

Anexo C - Código

; Periférico de

Entrada-----

PER_EN EQU 1A0H ; Endereço do periférico de entrada

NPEPE EQU 1B0H ;
Número do PEPE a verificar

PalavraPasseVerificar EQU 235H ; Palavra-passe
escrita pelo utilizador para verificação

; Periféricos de

Saída-----

Display_beginning EQU 200H ; Endereço
de início do display de 7 segmentos

Display_end EQU 26FH ; Endereço de fim do display de 7
segmentos

;Opcoes-----

; Menu Inicial

OpcaoComprar EQU 1 ;

Opção para comprar

OpcaoUsarCartao EQU 2 ; Opção
para usar cartão

OpcaoStock EQU 3 ; Opção
para ver o stock

; Menu Comprar

OpcaoEst1 EQU 1 ;

Opção para comprar bilhete da estação 1

OpcaoEst2 EQU 2 ;

Opção para comprar bilhete da estação 2

OpcaoEst3 EQU 3 ;

Opção para comprar bilhete da estação 3

OpcaoEst4 EQU 4		;
Opção para comprar bilhete da estação 4		
OpcaoEst5 EQU 5		;
Opção para comprar bilhete da estação 5		
OpcaoCancelarCompra EQU 6		; Opção
para cancelar a compra		
; Menu Preco Total		
OpcaoContinuarPrecoTotal EQU 1		; Opção para
continuar		
OpcaoCancelarPrecoTotal EQU 2		; Opção para
cancelar		
; Menu Inserir Dinheiro 1		
Opcao10CentimosInserir1 EQU 1		; Opção para
inserir 10 cêntimos		
Opcao20CentimosInserir1 EQU 2		; Opção para
inserir 20 cêntimos		
Opcao50CentimosInserir1 EQU 3		; Opção para
inserir 50 cêntimos		
OpcaoSeguinteInserir1 EQU 5		; Opção para
avançar		
OpcaoCancelarInserir1 EQU 7		; Opção para
cancelar		
; Menu Inserir Dinheiro 2		
Opcao1EuroInserir2 EQU 1		; Opção para
inserir 1 euro		
Opcao2EuroInserir2 EQU 2		; Opção para
inserir 2 euros		
Opcao5EurosInserir2 EQU 3		; Opção para
inserir 5 euros		
OpcaoSeguinteInserir2 EQU 5		; Opção para
avançar		
OpcaoCancelarInserir2 EQU 7		; Opção para
cancelar		
; Menu Inserir Dinheiro 3		
Opcao10EuroInserir3 EQU 1		; Opção para
inserir 10 euros		
Opcao20EuroInserir3 EQU 2		; Opção para
inserir 20 euros		
OpcaoSeguinteInserir3 EQU 5		; Opção para
avançar		

OpcaoCancelarInserir3 EQU 7 cancelar	; Opção para
; Menu Inserir Dinheiro	
OpcaoContinuarInserirDinheiro EQU 1	; Opção para continuar
; Menu Talao	
OpcaoContinuarTalao EQU 1	; Opção
para continuar	
; Menu Primeira Compra	
OpcaoContinuarPrimeiraCompra EQU 1	; Opção para
continuar	
; Menu Troco Devolvido	
OpcaoSeguinteTrocoDevolvido EQU 5	; Opção para
avançar	
OpcaoSairTrocoDevolvido EQU 7	; Opção para sair
; Menu Usar Cartão	
OpcaoContinuarCartao EQU 1	; Opção
para continuar	
OpcaoCancelarCartao EQU 5	; Opção
para cancelar	
; Menu Verificar Stock	
OpcaoConsultarVerificarStock EQU 1	; Opção para
consultar o stock	
OpcaoCancelarVerificarStock EQU 7	; Opção para
cancelar	
; Menu Consultar Stock	
OpcaoSeguinteStock EQU 5	; Opção para
avançar	
OpcaoSairStock EQU 7	; Opção
para sair	
; Menu Escolher Bilhete	
OpcaoSimEscolherBilhete EQU 1	; Opção para
escolher bilhete	
OpcaoNaoEscolherBilhete EQU 2	; Opção para não
escolher bilhete	
; Menu Erro	

OpcaoVoltarErro EQU 7 ; Opção
para voltar

; Menu Sem Troco
OpcaoContinuarSemTroco EQU 7 ; Opção para
continuar

; Menu Obrigado Compra
OpcaoContinuarObrigadoCompra EQU 7 ; Opção para
continuar

; Menu NPEPE Errado
OpcaoVoltarNPEPEErrado EQU 7 ; Opção para
voltar

; Menu PEPE
OpcaoComprarPEPE EQU 1 ; Opção
para comprar
OpcaoRecarregarPEPE EQU 2 ; Opção
para recarregar
OpcaoVoltarPEPE EQU 7 ; Opção
para voltar

; Menu Saldo Insuficiente PEPE
OpcaoVoltarSaldoInsuficientePEPE EQU 7 ; Opção para voltar

; Menu Talao PEPE
OpcaoContinuarTalaopepe EQU 1 ; Opção para
continuar

; Menu Inseriu Dinheiro PEPE
OpcaoContinuarInseriuDinheiroPEPE EQU 1 ; Opção para
continuar

;Constantes-----

CaracterVazio EQU 20H ; Caracter vazio

StackPointer EQU 1000H ; Endereço do Stack Pointer

;Endereços para escrita de
valores-----

; Stock
EndPrimeiroLugarStock EQU 22DH ; Endereço do primeiro lugar do display para escrever o valor no display do stock

; Devolução de Troco
EndPrimeiroLugarTroco EQU 22DH ; Endereço do primeiro lugar do display para escrever o valor no display de devolução do troco

; Preço Total
EndEscreverPrecoTotalCentimos EQU 239H ; Endereço do primeiro lugar do display para escrever os centimos do preço total no respetivo display
EndEscreverPrecoTotalEuros EQU 236H ; Endereço do primeiro lugar do display para escrever os euros preço total no respetivo display

; Dinheiro Inserido
EndDinheiroInseridoCentimos EQU 22EH ; Endereço do primeiro lugar do display para escrever os centimos do dinheiro inserido no respetivo display
EndDinheiroInseridoEuros EQU 22BH ; Endereço do primeiro lugar do display para escrever os euros do dinheiro inserido no respetivo display

; Talão
EndNumeroPEPETalao EQU 219H ; Endereço do primeiro lugar do display para escrever o número do PEPE no talão
EndPrecosCentimosTalao EQU 22EH ; Endereço do primeiro lugar do display para escrever os centimos do preco total no talão
EndPrecosEurosTalao EQU 22BH ; Endereço do primeiro lugar do display para escrever os euros do preco total no talão

; PEPE
EndNumeroPEPE EQU 209H ; Endereço do primeiro lugar do display para escrever o número do PEPE
EndSaldoPEPECentimos EQU 22DH ; Endereço do primeiro lugar do display para escrever os centimos do saldo do PEPE
EndSaldoPEPEEuros EQU 22AH ; Endereço do primeiro lugar do display para escrever os euros do saldo do PEPE

; Recarregar PEPE
EndSaldoPEPECentimosRecarregar EQU 23EH ; Endereço do primeiro lugar do display para escrever os centimos do saldo do PEPE a recarregar
EndSaldoPEPEEurosRecarregar EQU 23BH ; Endereço do primeiro lugar do display para escrever os euros do saldo do PEPE a recarregar

;Preços dos
bilhetes-----

BEstacao1Preco EQU 150 bilhete da estação 1	; Preço do
BEstacao2Preco EQU 250 bilhete da estação 2	; Preço do
BEstacao3Preco EQU 400 bilhete da estação 3	; Preço do
BEstacao4Preco EQU 550 bilhete da estação 4	; Preço do
BEstacao5Preco EQU 700 bilhete da estação 5	; Preço do

;Dinheiro-----

; Endereços do numero de moedas/notas para troco

TrocoMoedas10 EQU 300H
TrocoMoedas20 EQU 310H
TrocoMoedas50 EQU 320H
TrocoMoedas1 EQU 330H
TrocoMoedas2 EQU 340H
TrocoNotas5 EQU 350H
TrocoNotas10 EQU 360H
TrocoNotas20 EQU 370H

; Endereços dos números de moedas/notas em stock

StockMoedas10 EQU 380H
StockMoedas20 EQU 390H
StockMoedas50 EQU 3A0H
StockMoedas1 EQU 3B0H
StockMoedas2 EQU 3C0H
StockNotas5 EQU 3D0H
StockNotas10 EQU 3E0H
StockNotas20 EQU 3F0H

; Valores monetarios

Valor10Centimos EQU 10
Valor20Centimos EQU 20
Valor50Centimos EQU 50
Valor1Euro EQU 100
Valor2Euros EQU 200

Valor5Euros EQU 500
Valor10Euros EQU 1000
Valor20Euros EQU 2000

;Variáveis de

Compra-----

PrecoTotalCompra EQU 500H ; Preço
total da compra a ser feita

ValorInseridoCompra EQU 510H ; Valor total
inserido pelo utilizador durante o pagamento

ValorEmFaltaCompra EQU 520H ; Valor que
falta para completar o preço total da compra ($\text{PrecoTotalCompra} - \text{ValorInseridoCompra}$)

TrocoCompra EQU 530H ; Troco a
ser devolvido ao utilizador ($\text{ValorInseridoCompra} - \text{PrecoTotalCompra}$)

;PEPEs-----

BaseDeDadosPEPE EQU 1000H ; Endereço
inicial da base de dados dos PEPEs

FimBaseDeDadosPEPE EQU 1090H ; Endereço final da
base de dados dos PEPEs

NPepes EQU 10A0H ; Número
de PEPEs na base de dados (Máximo de 10 PEPEs)

NPEPEAtual EQU 10B0H ; Número
do PEPE atual a ser gerido

SaldoPEPEAtual EQU 10C0H ; Saldo do
PEPE atual a ser gerido

;Passe do

Stock-----

Place 100H

PalavraPasseStock:

;Palavra-passe para aceder ao stock (73 74 6F 63 6B)
String "stock"

;-----

; Menus

;-----

PLACE 2000H

Display_MenuInicial: ; Menu Inicial

```
String " MAQUINA VENDAS "  
String "  METRO  "  
String "      "  
String "1 - Comprar  "  
String "2 - Usar Cartao"  
String "3 - Stock  "  
String "      "
```

PLACE 2100H

Display_MenuComprar: ; Menu
Comprar

```
String " MENU ESTACAO "  
String "1-ESTACAO 1:1.50"  
String "2-ESTACAO 2:2.50"  
String "3-ESTACAO 3:4.00"  
String "4-ESTACAO 4:5.50"  
String "5-ESTACAO 5:7.00"  
String "6 - Cancelar  "
```

PLACE 2200H

Display_Talao: ;
Talão de compra

```
String " PEPE GERADO "  
String " 000  "  
String "Total:  .00 "  
String "Inserido: .00 "  
String "Troco:  .00 "  
String "      "  
String " 1 - Continuar  "
```

PLACE 2300H

Display_CodigoPEPE: ;

Menu para inserir o código do PEPE

```
String " INTRODUZA N. "  
String "   PEPE   "  
String "         "  
String "   XX   "  
String "         "  
String " 1 - Continuar "  
String " 5 - Cancelar  "
```

PLACE 2400H

Display_NPEPEErrado: ;

Mensagem de erro para N° PEPE inexistente

```
String "----- ERRO -----"  
String "         "  
String "   PEPE   "  
String " inexistente "  
String "         "  
String " 7 - Voltar   "  
String "----- ERRO -----"
```

PLACE 2500H

Display_MenuUsarPEPE: ; Menu

gestor do PEPE

```
String "   PEPE   "  
String "         "  
String " Saldo: 00.00 "  
String "         "  
String "1 - Comprar   "  
String "2 - Recarregar "  
String "7 - Voltar   "
```

PLACE 2600H

Display_ErroEspacoInsuficiente: ; Mensagem de

erro para espaço insuficiente na base de dados dos PEPEs

```
String "----- ERRO -----"  
String " ESPACO PARA "  
String "   PEPE   "  
String " INSUFICIENTE "  
String "         "
```

```
String " 7 - Voltar  "
String "----- ERRO -----"
```

Place 2700H

Display_SaldoInsuficientePEPE: ; Mensagem de
erro para saldo insuficiente no PEPE para a compra

```
String "----- ERRO -----"
String "                "
String "  SALDO PEPE  "
String " INSUFICIENTE "
String "                "
String " 7 - Voltar  "
String "----- ERRO -----"
```

Place 2800H

Display_VerificacaoStock: ; Menu para
entrada no stock da máquina

```
String "  STOCK  "
String "                "
String " PALAVRA-PASSE: "
String "  *****  "
String "                "
String "1 - Consultar  "
String "7 - Cancelar  "
```

Place 2900H

Display_InserirDinheiro1: ; Menu para
inserir dinheiro na máquina

```
String "  INERCAO  "
String " DE DINHEIRO "
String "1 - 10 Cent  "
String "2 - 20 Cent  "
String "3 - 50 Cent  "
String "5 - Seguinte  "
String "7 - Cancelar  "
```

Place 2A00H

Display_InserirDinheiro2: ; Menu para
inserir dinheiro na máquina

```
String "  INERCAO  "
String " DE DINHEIRO "
```

```
String "1 - 1 Euro    "
String "2 - 2 Euros   "
String "3 - 5 Euros   "
String "5 - Seguinte  "
String "7 - Cancelar  "
```

Place 2B00H

Display_InserirDinheiro3:

; Menu para

inserir dinheiro na máquina

```
String "  INERCAO  "
String " DE DINHEIRO "
String "1 - 10 Euros "
String "2 - 20 Euros "
String "           "
String "5 - Seguinte "
String "7 - Cancelar "
```

Place 2C00H

Display_PalavraPasseErrada:

; Mensagem de

erro para palavra-passe do stock incorreta

```
String " ---- ERRO ---- "
String " PALAVRA-PASSE "
String "  INERIDA  "
String " INCORRETA  "
String "           "
String " 7 - Voltar  "
String " ---- ERRO ---- "
```

Place 2D00H

Display_EscolheuBilhete:

; Menu para

escolher bilhete

```
String "-----"
String "ESCOLHEU BILHETE"
String "           "
String "DESEJA ESCOLHER "
String " OUTRA ESTACAO? "
String "1 - Sim 2 - Nao"
String "-----"
```

Place 2E00H

Display_PrecioTotal: ; Menu para

mostrar o preço total da compra

```
String "-----"
String " PRECO TOTAL  "
String "-----"
String "    .00    "
String "1 - Continuar  "
String "2 - Cancelar   "
String "-----"
```

Place 2F00H

Display_Erro: ;

Mensagem de erro para opção não disponível

```
String " ---- ERRO ---- "
String "                "
String "  OPCA0 NAO  "
String "  DISPONIVEL  "
String "                "
String " 7 - Voltar   "
String " ---- ERRO ---- "
```

Place 3000H

Display_InseriuDinheiro: ; Menu para

mostrar o dinheiro inserido

```
String "-----"
String "                "
String " INSERIU: .00 "
String " FALTA:  .00 "
String "                "
String "1 - Continuar  "
String "-----"
```

Place 3100H

Display_PrimeiraCompra: ; Menu para

primeira compra

```
String "  PARABENS  "
String "VIAGEM GRATUITA!"
String "                "
String "VALOR DA COMPRA "
String "  CARREGADO  "
String "  NO CARTAO   "
```

String "1 - Continuar "

Place 3200H

Display_TrocoDevolvido1: ; Menu para

mostrar troco devolvido

String " RETIRE O SEU "

String " TROCO! "

String " 10 Cent... "

String " 20 Cent... "

String " 50 Cent... "

String "5 - Seguinte "

String "7 - Sair "

Place 3300H

Display_TrocoDevolvido2: ; Menu para

mostrar troco devolvido

String " RETIRE O SEU "

String " TROCO! "

String " 1 Euro.... "

String " 2 Euros... "

String " 5 Euros... "

String "5 - Seguinte "

String "7 - Sair "

Place 3400H

Display_TrocoDevolvido3: ; Menu para

mostrar troco devolvido

String " RETIRE O SEU "

String " TROCO! "

String " 10 Euros.. "

String " 20 Euros.. "

String " "

String "5 - Seguinte "

String "7 - Sair "

Place 3500H

Display_SemTroco: ;

Mensagem de erro para falta de troco

String " ---- ERRO ---- "

String " NAO HA TROCO "


```
String " DISPONIVEL "
String "          "
String " 7 - Continuar "
String " Sem troco "
String " ---- ERRO ---- "
```

Place 3600H

Display_ObrigadoCompra: ; Menu para
agradecer a compra

```
String "          "
String " OBRIGADO "
String " PELA SUA "
String " COMPRA! "
String "          "
String " 7 - Continuar "
```

Place 3700H

Display_TalaoPEPE: ; Menu
para mostrar o talão do PEPE

```
String "COMPRA EFETUADA "
String " COM PEPE "
String "Total: 0.00 "
String "Saldo: 0.00 "
String "          "
String " 1 - Continuar "
String "          "
```

Place 3800H

Display_InseriuDinheiroPEPE: ; Menu para
mostrar o dinheiro inserido no PEPE

```
String "-----"
String "          "
String "          "
String " INSERIU: .00 "
String "          "
String "1 - Continuar "
String "-----"
```

Place 3900H

Display_ConsultarStock1: ; Menu para
consultar o stock

```
String " CONSULTA  "
String " STOCK    "
String " 10 Cent... "
String " 20 Cent... "
String " 50 Cent... "
String "5 - Seguinte "
String "7 - Sair    "
```

Place 3A00H

Display_ConsultarStock2: ; Menu para
consultar o stock

```
String " CONSULTA  "
String " STOCK    "
String " 1 Euro.... "
String " 2 Euros... "
String " 5 Euros... "
String "5 - Seguinte "
String "7 - Sair    "
```

Place 3B00H

Display_ConsultarStock3: ; Menu para
consultar o stock

```
String " CONSULTA  "
String " STOCK    "
String " 10 Euros.. "
String " 20 Euros.. "
String "           "
String "5 - Seguinte "
String "7 - Sair    "
```

```
;------
-----
;                               Código de Inicialização
;------
-----
```

Place 0000H ;
Começa o programa no endereço 0000H

Inicio:

```

        MOV R0, Principio                ; Coloca em R0 o endereço da etiqueta
Principio
        JMP R0                          ; Salta para a etiqueta Principio

;-----
;
;                                     Rotina
Principal
;-----
; Inicializa a máquina e mostrar o menu principal no display
;-----

Place 6000H                             ;
Começa as instruções no endereço 6000H

Principio:

        MOV SP, StackPointer            ; Inicializa o Stack
Pointer
        CALL LimpaDisplay                ; Limpa o
display
        CALL LimpaPerifericos            ; Limpa os
periféricos

LePER:

        MOV R0, PER_EN                  ;
R0 tem o endereço do botão PER_EN
        MOVB R1, [R0]                   ; Leitura
do periférico PER_EN
        CMP R1, 1                       ; Compara
o valor do botão PER_EN com 1
        JNE LePER                       ; Se
o valor do botão PER_EN for diferente de 1, repete a leitura
        CALL VerificaDinheiro            ; Verifica a
quantidade de cada tipo monetário no stock de modo a ter algum suficiente no início da
maquina

Ligado:

```

MOV R2, Display_MenuInicial ; R2 tem o endereço do menu inicial
CALL MostraDisplay ; Mostra o menu inicial no display
CALL LimpaPerifericos ; Limpa os periféricos

LeOpcao:

MOV R0, PER_EN ; R0 tem o endereço do botão PER_EN
MOVB R1, [R0] ; Leitura do periférico PER_EN
CMP R1, 0 ; Compara o valor do botão PER_EN com 0
JEQ LeOpcao ; Se o valor do botão for 0, refaz a leitura

CMP R1, OpcaoComprar ; Compara o valor do botão PER_EN com a opção de comprar
JEQ RotinaComprarIntermedio ; Se o valor do botão for igual à opção de comprar, salta para a etiqueta RotinaComprarIntermedio

CMP R1, OpcaoUsarCartao ; Compara o valor do botão PER_EN com a opção de usar cartão
JEQ RotinaUsarCartaoIntermedio ; Se o valor do botão for igual à opção de usar cartão, salta para a etiqueta RotinaUsarCartaoIntermedio

CMP R1, OpcaoStock ; Compara o valor do botão PER_EN com a opção de ver o stock
JEQ RotinaStockIntermedio ; Se o valor do botão for igual à opção de ver o stock, salta para a etiqueta RotinaStockIntermedio

CALL RotinaErro ; Se o valor do botão for diferente das opções disponíveis, salta para a etiqueta RotinaErro
JMP Ligado ; Salta para a etiqueta Ligado

RotinaComprarIntermedio: ; Etiqueta intermédia para a rotina de comprar que usa um JMP de modo a alcançar a etiqueta RotinaComprar
JMP ReiniciaVariaveisCompra ; Salta para a etiqueta ReiniciaVariaveisCompra

RotinaUsarCartaoIntermedio: ; Etiqueta intermédia para a rotina de usar cartão que usa um JMP de modo a alcançar a etiqueta RotinaUsarCartao

```

    JMP RotinaUsarCartao                ; Salta para a
etiqueta RotinaUsarCartao

```

```

RotinaStockIntermedio:                ; Etiqueta
intermédia para a rotina de ver o stock que usa um JMP de modo a alcançar a etiqueta
RotinaStock

```

```

    JMP VerificacaoStock                ; Salta para a
etiqueta VerificacaoStock

```

```

;-----
-----

```

```

;                                     Rotina
Limpa Display

```

```

;-----
-----

```

```

; Limpa o display

```

```

;-----
-----

```

LimpaDisplay:

```

    ; Guardar os registos
    PUSH R0
    PUSH R1
    PUSH R2

```

```

    MOV R0, Display_beginning          ; R0 tem o
endereço de início do display

```

```

    MOV R1, Display_end                ;
R1 tem o endereço de fim do display

```

CicloLimpa:

```

    MOV R2, CaracterVazio              ; R2 tem o
valor do caracter vazio

```

```

    MOVB [R0], R2                      ;

```

Coloca o caracter vazio no display

```

    ADD R0, 1                          ;

```

Adiciona 1 ao endereço do display

```

    CMP R0, R1                         ;

```

Compara o endereço do display com o endereço de fim do display

```

    JLE CicloLimpa                    ; Se

```

o endereço do display for menor ou igual ao endereço de fim do display, repete o ciclo

```

; Restaurar os registos
POP R2
POP R1
POP R0
RET

```

```

;-----
;
; Rotina
Limpa Periféricos
;-----
; Limpa/Zera os periféricos
;-----

```

LimpaPerifericos:

```

; Guardar os registos
PUSH R0
PUSH R1

MOV R0, PER_EN ; R0 tem o
endereço do botão PER_ENT
MOV R1, NPEPE ;
R1 tem o número do PEPE a verificar

MOV R9, 0 ; R9 tem o
valor 0

MOVB [R0], R9 ; Coloca o
valor 0 no botão PER_EN
MOVB [R1], R9 ;
Coloca o valor 0 no número do PEPE a verificar

; Restaurar os registos
POP R1
POP R0
RET

```

```

;-----
;
; Rotina
Mostrar Display

```



```

;-----
;-----
; Mostra a mensagem de erro no display se opção não existir
;-----
;-----

```

RotinaErro:

```

MOV R2, Display_Erro ; R2 tem o
endereço da mensagem de erro
CALL MostraDisplay ;
Mostra a mensagem de erro no display
CALL LimpaPerifericos ; Limpa os
periféricos

```

LeOpcaoErro:

```

MOV R0, PER_EN ;
R0 tem o endereço do botão PER_EN
MOVB R1, [R0] ;
Leitura do periférico PER_EN
CMP R1, 0 ;
Compara o valor do botão PER_EN com 0
JEQ LeOpcaoErro ; Se o valor
do botão for 0, refaz a leitura

```

```

CMP R1, OpcaoVoltarErro ; Compara o valor
do botão PER_EN com a opção de voltar
JEQ Voltar ; Se
o valor do botão for igual à opção de voltar, salta para a etiqueta Voltar

```

```

CALL RotinaErro ; Se o valor
do botão for diferente da opção de voltar, salta para a etiqueta RotinaErro
JMP Ligado ;
Salta para a etiqueta Ligado

```

```

Voltar: ; Etiqueta
para voltar
RET

```

```

;-----
;-----
; Rotina
Verifica Dinheiro

```



```

;-----
;
; Verifica a quantidade de cada tipo monetário no stock de modo a ter algum suficiente no
início da maquina
;-----

```

VerificaDinheiro:

```

    ; Guardar os registos
    PUSH R0
    PUSH R1
    PUSH R2
    PUSH R3
    PUSH R4
    PUSH R5
    PUSH R6

    MOV R0, StockMoedas10 ; R0 tem o
endereço do stock de moedas de 10 centimos (Que são as primeiras na memória)
    MOV R1, StockNotas20 ; R1 tem o
endereço do stock de notas de 20 euros (Que são as ultimas na memória)
    MOV R5, 16 ;
R5 tem o valor de 16 para passar de endereço em endereço
    MOV R6, 10 ;
R6 tem o valor 10 para fins de adição

```

CicloVerificaDinheiro:

```

    MOV R2, [R0] ;
R2 tem a quantidade de dinheiro do valor monetário atual
    MOV R3, R6 ;
R3 tem o valor 10
    CMP R2, R3 ;
Compara a quantidade de dinheiro do valor monetário atual com 10
    JGE DinheiroSuficiente ; Se a
quantidade de dinheiro do valor monetário atual for maior ou igual a 10, salta para a etiqueta
DinheiroSuficiente
    MOV R4, R6 ;
R4 tem o valor 10 para obter a diferença entre a quantidade de dinheiro do valor monetário
atual e 10
    SUB R4, R2 ;
Subtrai a quantidade de dinheiro do valor monetário atual a 10

```

```

        ADD R2, R4
;
Adiciona a diferença entre a quantidade de dinheiro do valor monetário atual e 10 à
quantidade de dinheiro do valor monetário atual

```

```

        MOV [R0], R2
;
Atualiza a quantidade de dinheiro do valor monetário atual

```

DinheiroSuficiente:

```

        ADD R0, R5
;
Adiciona 16 ao endereço do valor monetário atual

```

```

        CMP R0, R1
;
Compara o endereço do valor monetário atual com o endereço do stock de notas de 20 euros
        JLE CicloVerificaDinheiro ; Se o endereço do
valor monetário atual for menor ou igual ao endereço do stock de notas de 20 euros, repete o
ciclo

```

```

; Restaurar os registos
POP R6
POP R5
POP R4
POP R3
POP R2
POP R1
POP R0
RET

```

```

;-----
-----
;
; Rotina
Reiniciar Variaveis Compra
;-----
-----
; Reinicia as variáveis de compra
;-----
-----

```

ReiniciarVarCompra:

```

; Guardar os registos
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4

```

PUSH R5	
PUSH R6	
PUSH R7	
PUSH R8	
MOV R0, PrecoTotalCompra	; R0 tem o
endereço do preço total da compra	
MOV R1, ValorInseridoCompra	; R1 tem o
endereço do valor total inserido pelo utilizador	
MOV R2, ValorEmFaltaCompra	; R2 tem o
endereço do preço em falta da compra	
MOV R3, TrocoCompra	;
R3 tem o endereço do troco a ser devolvido ao utilizador	
MOV R4, 0	;
R4 tem o valor 0	
MOV [R0], R4	;
Atualiza o preço total da compra para 0	
MOV [R1], R4	;
Atualiza o valor total inserido pelo utilizador para 0	
MOV [R2], R4	;
Atualiza o preço em falta da compra para 0	
MOV [R3], R4	;
Atualiza o troco a ser devolvido ao utilizador para 0	
MOV R0, TrocoMoedas10	; R0 tem o
endereço do número de moedas de 10 centimos para troco	
MOV R1, TrocoMoedas20	; R1 tem o
endereço do número de moedas de 20 centimos para troco	
MOV R2, TrocoMoedas50	; R2 tem o
endereço do número de moedas de 50 centimos para troco	
MOV R3, TrocoMoedas1	; R3 tem o
endereço do número de moedas de 1 euro para troco	
MOV R5, TrocoMoedas2	; R4 tem o
endereço do número de moedas de 2 euros para troco	
MOV R6, TrocoNotas5	;
R5 tem o endereço do número de notas de 5 euros para troco	
MOV R7, TrocoNotas10	; R6 tem o
endereço do número de notas de 10 euros para troco	
MOV R8, TrocoNotas20	; R7 tem o
endereço do número de notas de 20 euros para troco	

```

MOV [R0], R4 ;
Atualiza o número de moedas de 10 centimos para troco para 0
MOV [R1], R4 ;
Atualiza o número de moedas de 20 centimos para troco para 0
MOV [R2], R4 ;
Atualiza o número de moedas de 50 centimos para troco para 0
MOV [R3], R4 ;
Atualiza o número de moedas de 1 euro para troco para 0
MOV [R5], R4 ;
Atualiza o número de moedas de 2 euros para troco para 0
MOV [R6], R4 ;
Atualiza o número de notas de 5 euros para troco para 0
MOV [R7], R4 ;
Atualiza o número de notas de 10 euros para troco para 0
MOV [R8], R4 ;
Atualiza o número de notas de 20 euros para troco para 0

```

```

; Restaurar os registos
POP R8
POP R7
POP R6
POP R5
POP R4
POP R3
POP R2
POP R1
POP R0
RET

```

```

;-----
-----
;
; Rotina
Escrever Display
;-----
-----
; Escreve no display uma certa quantidade de valores, linha a linha na mesma coluna
;-----
-----
; R8 -> Valor a escrever
; R9 -> Endereço a escrever
;-----
-----

```

EscreveDisplay:

```

; Guardar os registos
PUSH R0
PUSH R1
PUSH R2
PUSH R3

MOV R1, R9 ;
R1 tem o endereço a escrever
MOV R2, 10 ;
R2 tem o valor 10 usado para a conta para passar de decimal para hexadecimal
MOV R3, 48 ;
R3 tem o valor 48 que é o valor do caracter 0 em ASCII

CicloEscreveDisplay:

MOV R0, R8 ;
R0 tem o valor a escrever
MOD R0, R2 ;
R0 tem o resto da divisão do valor a escrever por 10
ADD R0, R3 ;
Adiciona o valor do caracter 0 em ASCII ao valor a escrever
MOVB [R1], R0 ;
Escreve o valor a escrever no display
SUB R1, 1 ;
Subtrai 1 ao endereço a escrever de modo a para o lugar a esquerda se o valor a escrever tiver
mais que 1 dígito
DIV R8, R2 ;
Divide o valor a escrever por 10 para obter o próximo dígito
CMP R8, 0 ;
Compara o valor a escrever com 0 para saber se tem mais um dígito
JEQ FimCicloEscreveDisplay ; Se o valor
for 0 acaba o ciclo
JMP CicloEscreveDisplay ; Se o valor
não for 0 repete o ciclo

FimCicloEscreveDisplay:

; Restaurar os registos
POP R3
POP R2
POP R1
POP R0
RET

```

```

;-----
;
;
Escrever Preco
;-----
;
; Escreve o preço num certo endereço no display
;-----
;
; R7 -> Preço a escrever
;-----
;
;-----

```

EscreverPreco:

```

; Guardar os registos
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R8
PUSH R9

MOV R1, 100 ;
R1 tem o valor 100 para obter os euros e os cêntimos do preço a escrever

MOV R0, R7 ;
R0 tem o preço a escrever
DIV R0, R1 ;
R0 tem os euros desse preço

MOV R3, R7 ;
R3 tem o preço a escrever
MOD R3, R1 ;
R3 tem os cêntimos desse preço

```

CicloEscrevePreco:

```

MOV R9, EndEscreverPrecoTotalEuros ; R9 tem o
endereço a escrever os euros do preço
MOV R8, R0 ;
R8 tem os euros do preço

```

```

CALL EscreveDisplay ;
Chama a rotina EscreveDisplay, que escreve os euros do preço no display
MOV R9, EndEscreverPrecoTotalCentimos ; R9 tem o endereço a
escrever os cêntimos do preço
MOV R8, R3 ;
R8 tem os cêntimos do preço
CALL EscreveDisplay ;
Chama a rotina EscreveDisplay, que escreve os cêntimos do preço no display

```

```

; Restaurar os registos
POP R9
POP R8
POP R3
POP R2
POP R1
POP R0
RET

```

```

;-----
;
; Rotina
Atualizar Preco Em Falta
;-----
;-----
; Atualiza e escreve no display o preço em falta da compra e o valor total inserido pelo
utilizador
;-----
;-----
; R5 -> Valor inserido pelo o utilizador
;-----
;-----

```

AtualizarPrecoEmFalta:

```

; Guardar os registos
PUSH R0
PUSH R1
PUSH R2
PUSH R6
PUSH R7

MOV R0, ValorInseridoCompra ; R0 tem o
endereço do valor total inserido pelo utilizador durante a compra

```

MOV R1, [R0]	;
R1 tem o valor total inserido pelo utilizador	
ADD R1, R5	;
Adiciona o valor inserido pelo utilizador ao valor total inserido	
MOV [R0], R1	;
Atualiza o valor total inserido pelo utilizador	
MOV R0, ValorEmFaltaCompra	; R0 tem o
endereço do preço em falta da compra	
MOV R1, [R0]	;
R1 tem o preço em falta da compra	
CMP R5, R1	;
Compara o valor inserido pelo utilizador com o preço em falta da compra	
JGE FoiPago	; Se
o valor inserido pelo utilizador for maior ou igual ao preço em falta da compra, salta para a etiqueta FoiPago	
SUB R1, R5	;
Subtrai o valor inserido pelo utilizador ao preço em falta da compra	
MOV [R0], R1	;
Atualiza o preço em falta da compra	
MOV R2, ValorInseridoCompra	; R2 tem o
endereço do valor total inserido pelo utilizador	
MOV R6, [R2]	;
R6 tem o valor total inserido pelo utilizador	
MOV R2, ValorEmFaltaCompra	; R2 tem o
endereço do preço em falta da compra	
MOV R7, [R2]	;
R7 tem o preço em falta da compra	
CALL EscreverPrecoAtualizado	; Chama a rotina
EscreverPrecoAtualizado, que escreve o preço em falta da compra e o valor inserido total atualizados no display	
JMP FimAtualizarPrecoEmFalta	; Salta para a
etiqueta FimAtualizarPrecoEmFalta	
FoiPago:	
MOV R1, 0	;
R1 tem o valor 0	


```

MOV [R0], R1 ;
Atualiza o preço em falta da compra para 0
MOV R10, 1 ;
R10 tem o valor 1 para sinalizar que o preço em falta foi pago

```

```

JMP FimAtualizarPrecoEmFalta ; Salta para a
etiqueta FimAtualizarPrecoEmFalta

```

FimAtualizarPrecoEmFalta:

```

; Restaurar os registos
POP R7
POP R6
POP R2
POP R1
POP R0
RET

;-----
-----
;
; Rotina
Adiciona Stock
;-----
-----
; Atualiza o stock de moedas/notas adicionando 1 à quantidade de dinheiro do valor
monetário inserido pelo utilizador
;-----
-----
; R5 -> Valor a adicionar ao stock / inserido pelo utilizador
;-----
-----

```

AdicionaStock:

```

; Guardar os registos
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4

MOV R0, StockMoedas10 ; R0 tem o
endereço do stock de moedas de 10 centimos (Que são as primeiras na memória)

```

MOV R1, StockNotas20 ; R1 tem o endereço do stock de notas de 20 euros (Que são as ultimas na memória)

MOV R3, 16 ;
R3 tem o valor de 16 para passar de linha em linha de endereços

CicloAtualizaStock:

MOV R4, Valor10Centimos ; R4 tem o valor de 10 centimos

CMP R4, R5 ;
Compara o valor a adicionar ao stock com o valor de 10 centimos

JEQ AtualizarStock ; Se o valor a adicionar ao stock for igual ao valor de 10 centimos, salta para a etiqueta AtualizarStock

ADD R0, R3 ;
Adiciona 16 ao endereço do valor monetário atual

MOV R4, Valor20Centimos ; R4 tem o valor de 20 centimos

CMP R4, R5 ;
Compara o valor a adicionar ao stock com o valor de 20 centimos

JEQ AtualizarStock ; Se o valor a adicionar ao stock for igual ao valor de 20 centimos, salta para a etiqueta AtualizarStock

ADD R0, R3 ;
Adiciona 16 ao endereço do valor monetário atual

MOV R4, Valor50Centimos ; R4 tem o valor de 50 centimos

CMP R4, R5 ;
Compara o valor a adicionar ao stock com o valor de 50 centimos

JEQ AtualizarStock ; Se o valor a adicionar ao stock for igual ao valor de 50 centimos, salta para a etiqueta AtualizarStock

ADD R0, R3 ;
Adiciona 16 ao endereço do valor monetário atual

MOV R4, Valor1Euro ;
R4 tem o valor de 1 euro

CMP R4, R5 ;
Compara o valor a adicionar ao stock com o valor de 1 euro

JEQ AtualizarStock ; Se o valor a adicionar ao stock for igual ao valor de 1 euro, salta para a etiqueta AtualizarStock

ADD R0, R3	;
Adiciona 16 ao endereço do valor monetário atual	
MOV R4, Valor2Euros	; R4 tem o
valor de 2 euros	
CMP R4, R5	;
Compara o valor a adicionar ao stock com o valor de 2 euros	
JEQ AtualizarStock	; Se o valor
a adicionar ao stock for igual ao valor de 2 euros, salta para a etiqueta AtualizarStock	
ADD R0, R3	;
Adiciona 16 ao endereço do valor monetário atual	
MOV R4, Valor5Euros	; R4 tem o
valor de 5 euros	
CMP R4, R5	;
Compara o valor a adicionar ao stock com o valor de 5 euros	
JEQ AtualizarStock	; Se o valor
a adicionar ao stock for igual ao valor de 5 euros, salta para a etiqueta AtualizarStock	
ADD R0, R3	;
Adiciona 16 ao endereço do valor monetário atual	
MOV R4, Valor10Euros	; R4 tem o
valor de 10 euros	
CMP R4, R5	;
Compara o valor a adicionar ao stock com o valor de 10 euros	
JEQ AtualizarStock	; Se o valor
a adicionar ao stock for igual ao valor de 10 euros, salta para a etiqueta AtualizarStock	
ADD R0, R3	;
Adiciona 16 ao endereço do valor monetário atual	
MOV R4, Valor20Euros	; R4 tem o
valor de 20 euros	
CMP R4, R5	;
Compara o valor a adicionar ao stock com o valor de 20 euros	
JEQ AtualizarStock	; Se o valor
a adicionar ao stock for igual ao valor de 20 euros, salta para a etiqueta AtualizarStock	
JMP FimAtualizaStock	; Salta para
a etiqueta FimAtualizaStock	

AtualizarStock:

```

MOV R2, [R0] ;
R2 tem a quantidade de dinheiro do valor monetário atual
ADD R2, 1 ;
Adiciona 1 à quantidade de dinheiro do valor monetário atual
MOV [R0], R2 ;
Atualiza a quantidade de dinheiro do valor monetário atual

```

FimAtualizaStock:

```

; Restaurar os registos
POP R4
POP R3
POP R2
POP R1
POP R0
RET

;-----
;
; Rotina
Escrever Preco Atualizado
;-----
; Escreve o preço total inserido pelo utilizador e o valor que falta para completar o preço total da compra
;-----
; R6 -> Valor total inserido pelo utilizador
; R7 -> Valor que falta para completar o preço total da compra
;-----

```

EscreverPrecoAtualizado:

```

; Guardar os registos
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4
PUSH R5
PUSH R8
PUSH R9

```

MOV R0, 100	;
R0 tem o valor 100 para obter os euros e os cêntimos	
MOV R3, EndDinheiroInseridoEuros	; R3 tem o
endereço a escrever os euros	
MOV R4, EndDinheiroInseridoCentimos	; R4 tem o
endereço a escrever os cêntimos	
MOV R5, 16	;
R5 tem o valor 16 para passar de linha em linha	
CicloEscrevePrecoAtualizado:	
MOV R1, R6	;
R1 tem o valor total inserido pelo utilizador	
DIV R1, R0	;
R1 tem os euros do valor total inserido pelo utilizador	
MOV R2, R6	;
R2 tem o valor total inserido pelo utilizador	
MOD R2, R0	;
R2 tem os cêntimos do valor total inserido pelo utilizador	
MOV R9, R3	;
R9 tem o endereço a escrever os euros	
MOV R8, R1	;
R8 tem os euros do valor total inserido pelo utilizador	
CALL EscreveDisplay	;
Chama a rotina EscreveDisplay, que escreve os euros do valor total inserido pelo utilizador no display	
MOV R9, R4	;
R9 tem o endereço a escrever os cêntimos	
MOV R8, R2	;
R8 tem os cêntimos do valor total inserido pelo utilizador	
CALL EscreveDisplay	;
Chama a rotina EscreveDisplay, que escreve os cêntimos do valor total inserido pelo utilizador no display	
MOV R1, R3	;
R1 tem o endereço a escrever os euros	
MOV R2, R4	;
R2 tem o endereço a escrever os cêntimos	

ADD R1, R5	;
Adiciona 16 ao endereço a escrever os euros para passar para a proxima linha	
ADD R2, R5	;
Adiciona 16 ao endereço a escrever os cêntimos para passar para a proxima linha	
MOV R3, R7	;
R3 tem o valor que falta para completar o preço total da compra	
DIV R3, R0	;
R3 tem os euros do valor que falta para completar o preço total da compra	
MOV R4, R7	;
R4 tem o valor que falta para completar o preço total da compra	
MOD R4, R0	;
R4 tem os cêntimos do valor que falta para completar o preço total da compra	
MOV R9, R1	;
R9 tem o endereço a escrever os euros	
MOV R8, R3	;
R8 tem os euros do valor que falta para completar o preço total da compra	
CALL EscreveDisplay	;
Chama a rotina EscreveDisplay, que escreve os euros do valor que falta para completar o preço total da compra no display	
MOV R9, R2	;
R9 tem o endereço a escrever os cêntimos	
MOV R8, R4	;
R8 tem os cêntimos do valor que falta para completar o preço total da compra	
CALL EscreveDisplay	;
Chama a rotina EscreveDisplay, que escreve os cêntimos do valor que falta para completar o preço total da compra no display	
; Restaurar os registos	
POP R9	
POP R8	
POP R5	
POP R4	
POP R3	
POP R2	
POP R1	
POP R0	
RET	

```

;-----
;
;
;
;-----
;
; Escreve os valores do número do PEPE criado, do preço total da compra, valor total inserido
pelo utilizador e troco no talão
;-----
;
;-----

```

EscreverTalao:

```

; Guardar os registos
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4
PUSH R5
PUSH R6
PUSH R7
PUSH R8
PUSH R9

MOV R0, 16 ;
R0 tem o valor 16 para passar de endereço em endereço
MOV R1, 100 ;
R1 tem o valor 100 para obter os euros e os cêntimos

MOV R2, EndNumeroPEPETalao ; R2 tem o
endereço a escrever o número do PEPE criado
MOV R3, EndPrecosEurosTalao ; R3 tem o
primeiro endereço a escrever os euros dos preços
MOV R4, EndPrecosCentimosTalao ; R4 tem o
primeiro endereço a escrever os cêntimos dos preços

; Escrever o número do PEPE
MOV R5, NPepes ;
R5 tem o endereço do número de PEPEs
MOV R8, [R5] ;
R8 tem o número de PEPEs

```

```

MOV R9, R2 ;
R9 tem o endereço a escrever o número do PEPE criado
CALL EscreveDisplay ;
Chama a rotina EscreveDisplay, que escreve o número do PEPE criado no display

; Escrever o preço total
MOV R5, PrecoTotalCompra ; R5 tem o
endereço do preço total da compra
MOV R6, [R5] ;
R6 tem o preço total da compra
DIV R6, R1 ;
R6 tem os euros do preco total
MOV R7, [R5] ;
R7 tem o preço total da compra
MOD R7, R1 ;
R7 tem os cêntimos do preco total

MOV R9, R3 ;
R9 tem o endereço a escrever os euros do preço total
MOV R8, R6 ;
R8 tem os euros do preço total
CALL EscreveDisplay ;
Chama a rotina EscreveDisplay, que escreve os euros do preço total no display
MOV R9, R4 ;
R9 tem o endereço a escrever os cêntimos do preço total
MOV R8, R7 ;
R8 tem os cêntimos do preço total
CALL EscreveDisplay ;
Chama a rotina EscreveDisplay, que escreve os cêntimos do preço total no display

MOV R6, R3 ;
R6 tem o endereço a escrever os euros dos preços
MOV R7, R4 ;
R7 tem o endereço a escrever os cêntimos dos preços
ADD R6, R0 ;
Adiciona 16 ao endereço a escrever os euros dos preços para passar para o próximo preço a
escrever que neste caso é o valor total inserido
ADD R7, R0 ;
Adiciona 16 ao endereço a escrever os cêntimos dos preços para passar para o próximo preço
a escrever que neste caso é o valor total inserido

; Escrever o valor inserido
MOV R5, ValorInseridoCompra ; R5 tem o
endereço do valor total inserido pelo utilizador

```


MOV R3, [R5]	;
R3 tem o valor total inserido pelo utilizador	
DIV R3, R1	;
R3 tem os euros do valor total inserido	
MOV R4, [R5]	;
R4 tem o valor total inserido pelo utilizador	
MOD R4, R1	;
R4 tem os cêntimos do valor total inserido	
MOV R9, R6	;
R9 tem o endereço a escrever os euros do valor total inserido	
MOV R8, R3	;
R8 tem os euros do valor total inserido	
CALL EscreveDisplay	;
Chama a rotina EscreveDisplay, que escreve os euros do valor total inserido no display	
MOV R9, R7	;
R9 tem o endereço a escrever os cêntimos do valor total inserido	
MOV R8, R4	;
R8 tem os cêntimos do valor total inserido	
CALL EscreveDisplay	;
Chama a rotina EscreveDisplay, que escreve os cêntimos do valor total inserido no display	
MOV R3, R6	;
R3 tem o endereço a escrever os euros dos preços	
MOV R4, R7	;
R4 tem o endereço a escrever os cêntimos dos preços	
ADD R3, R0	;
Adiciona 16 ao endereço a escrever os euros dos preços para passar para o próximo preço a escrever que neste caso é o troco	
ADD R4, R0	;
Adiciona 16 ao endereço a escrever os cêntimos dos preços para passar para o próximo preço a escrever que neste caso é o troco	
; Escrever o troco	
MOV R5, TrocoCompra	;
R5 tem o endereço do troco da compra	
MOV R6, [R5]	;
R6 tem o troco da compra	
DIV R6, R1	;
R6 tem os euros do troco	
MOV R7, [R5]	;
R7 tem o troco da compra	

```

MOD R7, R1
R7 tem os cêntimos do troco

```

```

MOV R9, R3
R9 tem o endereço a escrever os euros do troco

```

```

MOV R8, R6
R8 tem os euros do troco

```

```

CALL EscreveDisplay
Chama a rotina EscreveDisplay, que escreve os euros do troco no display

```

```

MOV R9, R4
R9 tem o endereço a escrever os cêntimos do troco

```

```

MOV R8, R7
R8 tem os cêntimos do troco

```

```

CALL EscreveDisplay
Chama a rotina EscreveDisplay, que escreve os cêntimos do troco no display

```

```

; Restaurar os registos

```

```

POP R9

```

```

POP R8

```

```

POP R7

```

```

POP R6

```

```

POP R5

```

```

POP R4

```

```

POP R3

```

```

POP R2

```

```

POP R1

```

```

POP R0

```

```

RET

```

```

;-----

```

```

-----

```

```

;
Rotina Criar

```

```

PEPE

```

```

;-----

```

```

-----

```

```

; Cria um novo PEPE guardando esse na base de dados dos PEPEs

```

```

;-----

```

```

-----

```

```

CriarPEPE:

```

```

; Guardar os registos

```

```

PUSH R0

```

```

PUSH R1

```

PUSH R2	
PUSH R3	
PUSH R4	
PUSH R5	
PUSH R6	
MOV R0, NPepes	;
R0 tem o endereço do número de PEPEs	
MOV R1, [R0]	;
R1 tem o número de PEPEs	
MOV R2, BaseDeDadosPEPE	;
R2 tem o endereço do início da base de dados dos PEPEs	
MOV R4, 16	;
R4 tem o valor 16 para passar de linha em linha	
ADD R1, 1	;
Adiciona 1 ao número de PEPEs	
MOV [R0], R1	;
Atualiza o número de PEPEs	
MOV R11, R1	;
R11 tem o número do PEPE a ser criado para ser carregado em outra rotina com o valor dos bilhetes depois da primeira compra	
SUB R2, R4	;
Subtrai 16 ao endereço da base de dados dos PEPEs para fins de ciclo de modo que ao adicionar começar no primeiro lugar da base de dados	
MOV R5, 0	;
R5 tem o valor 0 para fins de contagem de linhas para colocar o novo PEPE no lugar correto da base de dados dos PEPEs	
MOV R6, 10	;
R6 tem o valor 10 para fins de comparação com o número do PEPE a ser criado, de modo a saber se já existem 10 PEPEs	
CicloCriarPEPE:	
CMP R1, R6	;
Compara o número do PEPE a ser criado com 10	
JGT MaximoCriado	; Se o
número do PEPE a ser criado for maior que 10 quer dizer que já não tem mais espaço, salta para a etiqueta MaximoCriado	

```

        ADD R5, 1                                ;
Adiciona 1 ao número de linhas para colocar o novo PEPE
        ADD R2, R4                                ;
Adiciona 16 ao endereço da base de dados dos PEPEs para passar para o próximo lugar da
base de dados
        CMP R5, R1                                ;
Compara o número de linhas para colocar o novo PEPE com o número do PEPE a ser criado
        JNE CicloCriarPEPE                        ; Se o
número de linhas para colocar o novo PEPE for diferente do número do PEPE a ser criado,
repete o ciclo

        MOV R3, R1                                ;
R3 tem o número do PEPE a ser criado
        MOV [R2], R3                              ;
Atualiza a base de dados dos PEPEs com o número do PEPE a ser criado

        JMP FimCicloCriarPEPE                    ; Salta para
a etiqueta FimCicloCriarPEPE

MaximoCriado:

        MOV R10, 1                                ;
R10 tem o valor 1 para sinalizar que não tem lugar para mais nenhum PEPE

        JMP FimCicloCriarPEPE                    ; Salta para
a etiqueta FimCicloCriarPEPE

FimCicloCriarPEPE:

        ; Restaurar os registos
        POP R6
        POP R5
        POP R4
        POP R3
        POP R2
        POP R1
        POP R0
        RET

;-----
-----
;
Atualizar Saldo PEPE
Rotina

```

```

;-----
;
;-----
; Atualiza o saldo do PEPE com o valor do bilhete
;-----
;
; R10 -> Valor a ser carregado no PEPE
; R11 -> Número do PEPE a ser carregado
;-----
;
;-----

```

AtualizarSaldoPEPE:

```

; Guardar os registos
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4

MOV R0, BaseDeDadosPEPE ;
R0 tem o endereço da base de dados dos PEPEs
MOV R1, 16 ;
R1 tem o valor 16 para passar de endereço em endereço
MOV R3, R10 ;
R3 tem o valor a ser carregado no PEPE
MOV R4, R11 ;
R4 tem o número do PEPE a ser carregado

SUB R0, R1 ;
Subtrai 16 ao endereço da base de dados dos PEPEs para fins de ciclo de modo que ao
adicionar começar no primeiro lugar da base de dados dos PEPEs

```

CicloAtualizarSaldoPEPE:

```

ADD R0, R1 ;
Adiciona 16 ao endereço da base de dados dos PEPEs para passar para o próximo lugar da
base de dados
MOV R2, [R0] ;
R2 tem o número do PEPE na posição atual da base de dados dos PEPEs
CMP R2, R4 ;
Compara o número do PEPE na posição atual da base de dados dos PEPEs com o número do
PEPE a ser carregado

```

JNE CicloAtualizarSaldoPEPE ; Se o
 número do PEPE na posição atual da base de dados dos PEPEs for diferente do número do
 PEPE a ser carregado, repete o ciclo

ADD R0, 4 ;
 Adiciona 4 ao endereço da base de dados dos PEPEs para passar para o lugar do valor
 monetario do PEPE
 MOV [R0], R3 ;
 Atualiza o valor monetario do PEPE

; Restaurar os registos
 POP R4
 POP R3
 POP R2
 POP R1
 POP R0
 RET

;-----

 ; Rotina
 Devolve Troco
 ;-----

 ; Devolve o troco da compra, retirando o valor do troco da compra do stock da máquina
 ;-----

DevolveTroco:

; Guardar os registos
 PUSH R0
 PUSH R1
 PUSH R2
 PUSH R3
 PUSH R4

 MOV R4, TrocoCompra ;
 R4 tem o endereço do troco da compra
 MOV R0, [R4] ;
 R0 tem o troco da compra

CicloDevolveTroco:

CMP R0, 0	;
Compara o troco da compra com 0	
JZ FimCicloDevolveTroco	; Se o troco
da compra for 0, salta para a etiqueta FimCicloDevolveTroco	
MOV R3, TrocoNotas20	; R3 tem o
endereço do troco de notas de 20 euros	
MOV R1, R0	;
R1 tem o troco da compra	
MOV R2, 2000	;
R2 tem o valor 2000 para comparar com o troco da compra (20 euros)	
SUB R1, R2	;
Subtrai 2000 a copia do troco da compra	
CMP R1, 0	;
Compara a copia do troco da compra com 0	
JGE Devolver	; Se a
copia do troco da compra for maior ou igual a 0, salta para a etiqueta Devolver	
MOV R3, TrocoNotas10	; R3 tem o
endereço do troco de notas de 10 euros	
MOV R1, R0	;
R1 tem o troco da compra	
MOV R2, 1000	;
R2 tem o valor 1000 para comparar com o troco da compra (10 euros)	
SUB R1, R2	;
Subtrai 1000 a copia do troco da compra	
CMP R1, 0	;
Compara a copia do troco da compra com 0	
JGE Devolver	; Se a
copia do troco da compra for maior ou igual a 0, salta para a etiqueta Devolver	
MOV R3, TrocoNotas5	;
R3 tem o endereço do troco de notas de 5 euros	
MOV R1, R0	;
R1 tem o troco da compra	
MOV R2, 500	;
R2 tem o valor 500 para comparar com o troco da compra (5 euros)	
SUB R1, R2	;
Subtrai 500 a copia do troco da compra	
CMP R1, R2	;
Compara a copia do troco da compra com 0	

JGE Devolver	; Se a
copia do troco da compra for maior ou igual a 0, salta para a etiqueta Devolver	
MOV R3, TrocoMoedas2	; R3 tem o
endereço do troco de moedas de 2 euros	
MOV R1, R0	;
R1 tem o troco da compra	
MOV R2, 200	;
R2 tem o valor 200 para comparar com o troco da compra (2 euros)	
SUB R1, R2	;
Subtrai 200 a copia do troco da compra	
CMP R1, 0	;
Compara a copia do troco da compra com 0	
JGE Devolver	; Se a
copia do troco da compra for maior ou igual a 0, salta para a etiqueta Devolver	
MOV R3, TrocoMoedas1	; R3 tem o
endereço do troco de moedas de 1 euro	
MOV R1, R0	;
R1 tem o troco da compra	
MOV R2, 100	;
R2 tem o valor 100 para comparar com o troco da compra (1 euro)	
SUB R1, R2	;
Subtrai 100 a copia do troco da compra	
CMP R1, 0	;
Compara a copia do troco da compra com 0	
JGE Devolver	; Se a
copia do troco da compra for maior ou igual a 0, salta para a etiqueta Devolver	
MOV R3, TrocoMoedas50	; R3 tem o
endereço do troco de moedas de 50 cêntimos	
MOV R1, R0	;
R1 tem o troco da compra	
MOV R2, 50	;
R2 tem o valor 50 para comparar com o troco da compra (50 cêntimos)	
SUB R1, R2	;
Subtrai 50 a copia do troco da compra	
CMP R1, 0	;
Compara a copia do troco da compra com 0	
JGE Devolver	; Se a
copia do troco da compra for maior ou igual a 0, salta para a etiqueta Devolver	

MOV R3, TrocoMoedas20 ; R3 tem o
endereço do troco de moedas de 20 centimos

MOV R1, R0 ;
R1 tem o troco da compra
MOV R2, 20 ;
R2 tem o valor 20 para comparar com o troco da compra (20 centimos)
SUB R1, R2 ;
Subtrai 20 a copia do troco da compra
CMP R1, 0 ;
Compara a copia do troco da compra com 0
JGE Devolver ; Se a
copia do troco da compra for maior ou igual a 0, salta para a etiqueta Devolver

MOV R3, TrocoMoedas10 ; R3 tem o
endereço do troco de moedas de 10 centimos

MOV R1, R0 ;
R1 tem o troco da compra
MOV R2, 10 ;
R2 tem o valor 10 para comparar com o troco da compra (10 centimos)
SUB R1, R2 ;
Subtrai 10 a copia do troco da compra
CMP R1, 0 ;
Compara a copia do troco da compra com 0
JEQ CicloDevolveTroco ; Se a
copia do troco da compra for igual a 0, salta para a etiqueta CicloDevolveTroco

Devolver:

SUB R0, R2 ; Subtrai o
valor do valor monetário devolvido ao troco da compra
MOV R2, [R3] ;
R2 tem o valor atual da quantidade de notas ou moedas desse tipo a devolver
ADD R2, 1 ;
Adiciona 1 ao valor atual da quantidade de notas ou moedas desse tipo a devolver
MOV [R3], R2 ;
Atualiza o valor da quantidade de notas ou moedas desse tipo a devolver
JMP CicloDevolveTroco ; Repete o
ciclo

FimCicloDevolveTroco:

```

; Restaurar os registos
POP R4
POP R3
POP R2
POP R1
POP R0
RET

```

```

;-----
;
;
Retira Stock
;-----
;
; Retira do stock de moedas/notas o troco a dar ao utilizador
;-----
;-----

```

RetiraStock:

```

; Guardar os registos
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4
PUSH R5

MOV R0, TrocoMoedas10 ; R0 tem o
endereço das moedas de 10 centimos a dar como troco (Que são as primeiras na memória)
MOV R5, TrocoNotas20 ; R5 tem o
endereço das notas de 20 euros a dar como troco (Que são as ultimas na memória)
MOV R1, StockMoedas10 ; R1 tem o
endereço do stock de moedas de 10 centimos (Que são as primeiras na memória)
MOV R2, 16 ;
R2 tem o valor 16 para passar de endereço em endereço

```

CicloRetiraStock:

```

MOV R3, [R0] ;
R3 tem a quantidade de moedas de um certo valor a dar como troco
MOV R4, [R1] ;
R4 tem a quantidade de moedas de um certo valor no stock

```

CMP R3, R4 ;
Compara a quantidade de moedas de um certo valor a dar como troco com a quantidade de moedas de um certo valor no stock

JGT ErroSemTroco ; Se a
quantidade de moedas de um certo valor a dar como troco for maior que a quantidade de moedas de um certo valor no stock, salta para a etiqueta ErroSemTroco

SUB R4, R3 ;
Subtrai a quantidade de moedas de um certo valor a dar como troco à quantidade de moedas de um certo valor no stock

MOV [R1], R4 ;
Atualiza a quantidade de moedas de um certo valor no stock

ADD R0, R2 ;
Adiciona 16 ao endereço atual a dar como troco para passar para o próximo valor monetário

ADD R1, R2 ;
Adiciona 16 ao endereço atual no stock para passar para o próximo valor monetário

CMP R0, R5 ;
Compara o endereço atual a dar como troco com o endereço das notas de 20 euros

JNE CicloRetiraStock ; Se o endereço
atual a dar como troco for diferente do endereço das notas de 20 euros, repete o ciclo

JMP FimRetiraStock ; Salta para
a etiqueta FimRetiraStock

ErroSemTroco:

MOV R10, 1 ;
R10 tem o valor 1 para sinalizar que não há troco

FimRetiraStock:

POP R5
POP R4
POP R3
POP R2
POP R1
POP R0
RET

;
Rotina
Escrever Troco

```

;-----
; Escreve a quantidade de cada valor monetário a dar como troco no display
;-----
; R6 -> Quantidade de valores a escrever
; R7 -> Endereço do primeiro valor a escrever
;-----

```

EscreverTroco:

```

    ; Guardar os registos
    PUSH R0
    PUSH R1
    PUSH R2
    PUSH R3
    PUSH R8
    PUSH R9

    MOV R0, 1 ;
R0 é o contador de valores escritos e começa a 1
    MOV R1, R6 ;
R1 tem a quantidade de valores a escrever
    MOV R2, R7 ;
R2 tem o endereço do primeiro valor a escrever
    MOV R3, 16 ;
R3 tem o valor 16 que é o número de bytes entre cada valor a escrever
    MOV R9, EndPrimeiroLugarTroco ; R9 tem o
endereço do primeiro lugar do display a escrever

```

CicloEscreveTroco:

```

    MOV R8, [R2] ;
R8 tem a quantidade do valor monetário atual
    CALL EscreveDisplay ;
Chama a rotina EscreveDisplay, que escreve a quantidade do valor monetário atual (valor em
R8) no display (endereço em R9)
    ADD R0, 1 ;
Adiciona 1 ao contador de valores escritos
    CMP R0, R1 ;
Compara o contador de valores escritos com a quantidade de valores a escrever

```

```

JGT FimCicloEscreveTroco ; Se o contador de
valores escritos for maior que a quantidade de valores a escrever, salta para a etiqueta
FimCicloEscreveTroco
ADD R2, R3 ;
Adiciona 16 ao endereço do valor monetário atual de modo a passar para o próximo valor
monetário
ADD R9, R3 ;
Adiciona 16 ao endereço do display de modo a passar para o próximo lugar do display
JMP CicloEscreveTroco ; Repete o
ciclo

```

FimCicloEscreveTroco:

```

; Restaurar os registos
POP R9
POP R8
POP R3
POP R2
POP R1
POP R0
RET

;-----
;-----
; Rotina
Verifica PEPE
;-----
;-----
; Verifica se o PEPE introduzido pelo utilizador existe na base de dados dos PEPEs
;-----
;-----

```

VerificaPEPE:

```

; Guardar os registos
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4
PUSH R5
PUSH R6

```

MOV R0, BaseDeDadosPEPE	;
R0 tem o endereço da base de dados dos PEPEs	
MOV R1, 16	;
R1 tem o valor 16 para passar de endereço em endereço	
MOV R3, NPEPE	;
R3 tem o endereço do PEPE introduzido pelo utilizador	
MOVB R4, [R3]	;
R4 tem o número do PEPE introduzido pelo utilizador	
CMP R4, 0	;
Compara o número do PEPE introduzido pelo utilizador com 0	
JEQ NaoEncontrado	; Se o
número do PEPE introduzido pelo utilizador for 0, salta para a etiqueta NaoEncontrado	
MOV R5, 0	;
R5 tem o valor 0 para fins de contagem de linhas para verificar se o PEPE introduzido pelo utilizador existe na base de dados dos PEPEs	
MOV R6, 10	;
R6 tem o valor 10 para fins de comparação com o número de linhas para verificar se o PEPE introduzido pelo utilizador existe na base de dados dos PEPEs	
CicloVerificaPEPE:	
CMP R5, R6	;
Compara o número de linhas para verificar se o PEPE introduzido pelo utilizador existe na base de dados dos PEPEs com 10	
JEQ NaoEncontrado	; Se o
número de linhas para verificar se o PEPE introduzido pelo utilizador existe na base de dados dos PEPEs for igual a 10 quer dizer que o PEPE introduzido pelo utilizador não existe, salta para a etiqueta NaoEncontrado	
ADD R5, 1	;
Adiciona 1 ao número de linhas verificadas	
MOV R2, [R0]	;
R2 tem o número do PEPE atual	
ADD R0, R1	;
Adiciona 16 ao endereço da base de dados dos PEPEs para passar para o próximo lugar da base de dados	
CMP R2, R4	;
Compara o número do PEPE atual com o número do PEPE introduzido pelo utilizador	
JNE CicloVerificaPEPE	; Se o
número do PEPE atual for diferente do número do PEPE introduzido pelo utilizador, repete o ciclo	

JMP Encontrado ; Se
o número do PEPE atual for igual ao número do PEPE introduzido pelo utilizador, salta para
a etiqueta Encontrado

NaoEncontrado:

MOV R7, 0 ;
R7 tem o valor 0, que quer dizer que o PEPE introduzido pelo utilizador não existe
JMP FimCicloVerificaPEPE ; Salta para a
etiqueta FimCicloVerificaPEPE

Encontrado:

MOV R7, 1 ;
R7 tem o valor 1, que quer dizer que o PEPE introduzido pelo utilizador existe

MOV R3, NPEPEAtual ;
R1 tem o endereço do PEPE atual
MOV [R3], R2 ;
Atualiza o PEPE atual

SUB R0, R1 ;
Subtrai o endereço do PEPE para ir para o do atual
ADD R0, 4 ;
Adiciona 4 ao endereço da base de dados dos PEPEs para passar para o lugar do valor
monetario do PEPE

MOV R2, [R0] ;
R2 tem o valor monetario do PEPE
MOV R1, SaldoPEPEAtual ; R1 tem o
endereço do saldo do PEPE atual
MOV [R1], R2 ;
Atualiza o saldo do PEPE atual

JMP FimCicloVerificaPEPE ; Salta para a
etiqueta FimCicloVerificaPEPE

FimCicloVerificaPEPE:

; Restaurar os registos
POP R6
POP R5
POP R4
POP R3

```

POP R2
POP R1
POP R0
RET

```

```

;-----
;
; Rotina
Escrever Cartao
;-----
; Escreve o número e o saldo do PEPE no display
;-----

```

EscreverCartao:

```

; Guardar os registos
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4
PUSH R5
PUSH R6
PUSH R8
PUSH R9

MOV R0, EndNumeroPEPE ; R0 tem o
endereço a escrever o numero do PEPE no cartão
MOV R1, EndSaldoPEPECentimos ; R1 tem o
endereço a escrever os centimos do saldo do PEPE no cartão
MOV R6, EndSaldoPEPEEuros ; R6 tem o
endereço a escrever os euros do saldo do PEPE no cartão

MOV R2, 100 ;
R2 tem o valor 100 para obter os euros e cêntimos do saldo

```

CicloEscreverCartao:

```

MOV R3, NPEPEAtual ;
R3 tem o endereço do número do PEPE atual
MOV R8, [R3] ;
R8 tem o número do PEPE atual

```



```

MOV R9, R0 ;
R9 tem o endereço a escrever o numero do PEPE no cartão
CALL EscreveDisplay ;
Chama a rotina EscreveDisplay, que escreve o número do PEPE atual no cartão

MOV R3, SaldoPEPEAtual ; R3 tem o
endereço do saldo do PEPE atual
MOV R4, [R3] ;
R4 tem o saldo do PEPE atual
DIV R4, R2 ;
R4 tem os euros do saldo do PEPE atual
MOV R5, [R3] ;
R5 tem o saldo do PEPE atual
MOD R5, R2 ;
R5 tem os cêntimos do saldo do PEPE atual

MOV R9, R6 ;
R9 tem o endereço a escrever os euros do saldo do PEPE atual no cartão
MOV R8, R4 ;
R8 tem os euros do saldo do PEPE atual
CALL EscreveDisplay ;
Chama a rotina EscreveDisplay, que escreve os euros do saldo do PEPE atual no cartão
MOV R9, R1 ;
R9 tem o endereço a escrever os cêntimos do saldo do PEPE atual no cartão
MOV R8, R5 ;
R8 tem os cêntimos do saldo do PEPE atual
CALL EscreveDisplay ;
Chama a rotina EscreveDisplay, que escreve os cêntimos do saldo do PEPE atual no cartão

; Restaurar os registos
POP R9
POP R8
POP R6
POP R5
POP R4
POP R3
POP R2
POP R1
POP R0
RET

```


```

;
Escrever Talao PEPE
;-----
-----
; Escreve o preço da compra e o saldo do PEPE atual no talão da compra usando um PEPE
;-----
-----

```

EscreverTalaoPEPE:

```

; Guardar os registos
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4
PUSH R5
PUSH R6
PUSH R8
PUSH R9

MOV R0, 16
R0 tem o valor 16 para passar de endereço em endereço
MOV R1, 100
R1 tem o valor 100 para obter os euros e os centimos

MOV R2, EndPrecosEurosTalao
endereço a escrever os euros dos preços no talão
MOV R3, EndPrecosCentimosTalao
endereço a escrever os centimos dos preços no talão
; R2 tem o
; R3 tem o

; Escreve o preço da compra no talão
MOV R4, PrecoTotalCompra
endereço do preço total da compra
MOV R5, [R4]
R5 tem o preço total da compra
DIV R5, R1
R5 tem os euros do preço total da compra
MOV R6, [R4]
R6 tem o preço total da compra
MOD R6, R1
R6 tem os centimos do preço total da compra
; R4 tem o
;
;
;

```

```

MOV R9, R2 ;
R9 tem o endereço a escrever os euros do preço total da compra no talão
MOV R8, R5 ;
R8 tem os euros do preço total da compra
CALL EscreveDisplay ;
Chama a rotina EscreveDisplay, que escreve os euros do preço total da compra no talão
MOV R9, R3 ;
R9 tem o endereço a escrever os centimos do preço total da compra no talão
MOV R8, R6 ;
R8 tem os centimos do preço total da compra
CALL EscreveDisplay ;
Chama a rotina EscreveDisplay, que escreve os centimos do preço total da compra no talão

ADD R2, R0 ;
Adiciona 16 ao endereço a escrever os euros dos preços no talão para passar para o próximo
preço
ADD R3, R0 ;
Adiciona 16 ao endereço a escrever os centimos dos preços no talão para passar para o
próximo preço

; Escreve o valor do saldo do PEPE atual no talão
MOV R4, SaldoPEPEAtual ; R4 tem o
endereço do saldo do PEPE atual
MOV R5, [R4] ;
R5 tem o saldo do PEPE atual
DIV R5, R1 ;
R5 tem os euros do saldo do PEPE atual
MOV R6, [R4] ;
R6 tem o saldo do PEPE atual
MOD R6, R1 ;
R6 tem os centimos do saldo do PEPE atual

MOV R9, R2 ;
R9 tem o endereço a escrever os euros do saldo do PEPE atual no talão
MOV R8, R5 ;
R8 tem os euros do saldo do PEPE atual
CALL EscreveDisplay ;
Chama a rotina EscreveDisplay, que escreve os euros do saldo do PEPE atual no talão
MOV R9, R3 ;
R9 tem o endereço a escrever os centimos do saldo do PEPE atual no talão
MOV R8, R6 ;
R8 tem os centimos do saldo do PEPE atual
CALL EscreveDisplay ;
Chama a rotina EscreveDisplay, que escreve os centimos do saldo do PEPE atual no talão

```

```

; Restaurar os registos
POP R9
POP R8
POP R6
POP R5
POP R4
POP R3
POP R2
POP R1
POP R0
RET

```

```

;-----
;
; Rotina
Escrever Inserido PEPE
;-----
; Escreve o valor total inserido no recarregamento pelo o utilizador no PEPE no display
;-----
; R5 -> Valor inserido pelo o utilizador
;-----

```

EscreverInseridoPEPE:

```

; Guardar os registos
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4
PUSH R6
PUSH R8
PUSH R9

MOV R0, ValorInseridoCompra ; R0 tem o
endereço do valor total inserido no recarregamento do PEPE
MOV R6, [R0] ; R6 tem o
valor total inserido no recarregamento do PEPE
ADD R6, R5 ; Adiciona
o valor inserido pelo utilizador ao valor total inserido no recarregamento do PEPE

```

MOV [R0], R6	; Atualiza
o valor total inserido no recarregamento do PEPE	
MOV R0, 100	; R0 tem o
valor 100 para obter os euros e os cêntimos do saldo	
MOV R3, EndSaldoPEPECentimosRecarregar	; R3 tem o endereço a
escrever os cêntimos do dinheiro inserido no PEPE	
MOV R4, EndSaldoPEPEEurosRecarregar	; R4 tem o endereço a escrever os
euros do dinheiro inserido no PEPE	
CicloEscreverInseridoPEPE:	
MOV R1, R6	; R1 tem o
dinheiro inserido no PEPE	
DIV R1, R0	; R1 tem os
euros do dinheiro inserido no PEPE	
MOV R2, R6	; R2 tem o
dinheiro inserido no PEPE	
MOD R2, R0	; R2 tem os
cêntimos do dinheiro inserido no PEPE	
MOV R9, R4	; R9 tem o
endereço a escrever os euros do dinheiro inserido no PEPE	
MOV R8, R1	; R8 tem os
euros do dinheiro inserido no PEPE	
CALL EscreveDisplay	; Chama a
rotina EscreveDisplay, que escreve os euros do dinheiro inserido no PEPE	
MOV R9, R3	; R9 tem o
endereço a escrever os cêntimos do dinheiro inserido no PEPE	
MOV R8, R2	; R8 tem os
cêntimos do dinheiro inserido no PEPE	
CALL EscreveDisplay	; Chama a
rotina EscreveDisplay, que escreve os cêntimos do dinheiro inserido no PEPE	
; Restaurar os registos	
POP R9	
POP R8	
POP R6	
POP R4	
POP R3	
POP R2	
POP R1	
POP R0	
RET	

```

;-----
-----
;
Verifica Palavra-Passe
;-----
-----
; Verifica se a palavra-passe introduzida pelo utilizador está correta
;-----
-----

```

VerificaPalavraPasseStock:

```

; Guardar os registos
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4

MOV R0, PalavraPasseStock ; R0 tem o
endereço da palavra-passe para aceder ao stock
MOV R1, PalavraPasseVerificar ; R1 tem o
endereço da palavra-passe introduzida pelo utilizador

```

CicloVerificaPalavraPasseStock:

```

MOVB R2, [R0] ;
R2 tem o caracter da palavra-passe para aceder ao stock
MOVB R3, [R1] ;
R3 tem o caracter da palavra-passe introduzida pelo utilizador
CMP R2, R3 ;
Compara o caracter da palavra-passe para aceder ao stock com o caracter da palavra-passe
introduzida pelo utilizador
JNE PalavraPasseErrada ; Se os
caracteres forem diferentes, salta para a etiqueta PalavraPasseErrada
ADD R0, 1 ;
Adiciona 1 ao endereço da palavra-passe para aceder ao stock
ADD R1, 1 ;
Adiciona 1 ao endereço da palavra-passe introduzida pelo utilizador
MOVB R4, [R0] ;
R4 tem o próximo caracter da palavra-passe para aceder ao stock

```

```

        CMP R4, 0
;
Compara o próximo caracter da palavra-passe para aceder ao stock com 0
        JNE CicloVerificaPalavraPasseStock ; Se o próximo caracter da
palavra-passe para aceder ao stock não for 0, repete o ciclo
        JMP PalavraPasseCorreta ; Se o
próximo caracter da palavra-passe para aceder ao stock for 0, salta para a etiqueta
PalavraPasseCorreta

```

PalavraPasseErrada:

```

        MOV R6, 0
;
R6 tem o valor 0, que quer dizer que a palavra-passe está errada
        JMP FimCicloVerificaPalavraPasseStock

```

PalavraPasseCorreta:

```

        MOV R6, 1
;
R6 tem o valor 0, que quer dizer que a palavra-passe está errada
        JMP FimCicloVerificaPalavraPasseStock

```

FimCicloVerificaPalavraPasseStock:

```

; Restaurar os registos
POP R4
POP R3
POP R2
POP R1
POP R0
RET

```

```

;-----
-----
;
; Rotina
Escrever Stock
;-----
-----
; Escreve as quantidades de cada valor monetário do stock da maquina no display
;-----
-----
; R6 -> Quantidade de valores a escrever
; R7 -> Endereço do primeiro valor a escrever
;-----
-----

```

EscreverStock:

```
    ; Guardar os registos
    PUSH R0
    PUSH R1
    PUSH R2
    PUSH R3
    PUSH R8
    PUSH R9

    MOV R0, 1 ;
R0 é o contador de valores escritos e começa a 1
    MOV R1, R6 ;
R1 tem a quantidade de valores a escrever
    MOV R2, R7 ;
R2 tem o endereço do primeiro valor a escrever
    MOV R3, 16 ;
R3 tem o valor 16 que é o número de bytes entre cada valor a escrever
    MOV R9, EndPrimeiroLugarStock ; R9 tem o
endereço do primeiro lugar do display a escrever
```

CicloEscreveStock:

```
    MOV R8, [R2] ;
R8 tem a quantidade do valor monetário atual
    CALL EscreveDisplay ;
Chama a rotina EscreveDisplay, que escreve a quantidade do valor monetário atual (valor em
R8) no display (endereço em R9)
    ADD R0, 1 ;
Adiciona 1 ao contador de valores escritos
    CMP R0, R1 ;
Compara o contador de valores escritos com a quantidade de valores a escrever
    JGT FimCicloEscreveStock ; Se o contador de
valores escritos for maior que a quantidade de valores a escrever, salta para a etiqueta
FimCicloEscreveStock
    ADD R2, R3 ;
Adiciona 16 ao endereço do valor monetário atual de modo a passar para o próximo valor
monetário
    ADD R9, R3 ;
Adiciona 16 ao endereço do display de modo a passar para o próximo lugar do display
    JMP CicloEscreveStock ; Repete o
ciclo
```

FimCicloEscreveStock:


```

; Restaurar os registos
POP R9
POP R8
POP R3
POP R2
POP R1
POP R0
RET

```

```

;-----
;
; Etiqueta
Comprar
;-----

```

```

; Etiqueta que permite ao utilizador comprar bilhetes e consequentemente criar um cartão
PEPE
;-----

```

ReiniciaVariaveisCompra:

```

CALL ReiniciarVarCompra ; Chama a
rotina ReiniciarVarCompra, que reinicia as variáveis da compra
JMP RotinaComprar ; Salta para
a etiqueta RotinaComprar

```

```

;Escolha de
Bilhetes-----

```

RotinaComprar:

```

MOV R2, Display_MenuComprar ; R2 tem o
endereço do display do menu de compra
CALL MostraDisplay ; Mostra o
display do menu de compra
CALL LimpaPerifericos ; Limpa os
periféricos

```

LeOpcaoComprar:

```

MOV R0, PER_EN ;
R0 tem o endereço do periférico de entrada

```

MOVB R1, [R0]	; R1 tem o
valor do periférico de entrada	
CMP R1, 0	; Compara
o valor do periférico de entrada com 0	
JEQ LeOpcaoComprar	; Se
o valor do periférico de entrada for 0, repete o ciclo	
MOV R5, BEstacao1Preco	; R5 tem o
preço da estação 1	
CMP R1, OpcaoEst1	; Compara
a opção escolhida pelo utilizador com a opção da estação 1	
JEQ EscolhaBilhetes	; Se a
opção escolhida pelo utilizador for a opção da estação 1, salta para a etiqueta EscolhaBilhetes	
MOV R5, BEstacao2Preco	; R5 tem o
preço da estação 2	
CMP R1, OpcaoEst2	; Compara
a opção escolhida pelo utilizador com a opção da estação 2	
JEQ EscolhaBilhetes	; Se a
opção escolhida pelo utilizador for a opção da estação 2, salta para a etiqueta EscolhaBilhetes	
MOV R5, BEstacao3Preco	; R5 tem o
preço da estação 3	
CMP R1, OpcaoEst3	; Compara
a opção escolhida pelo utilizador com a opção da estação 3	
JEQ EscolhaBilhetes	; Se a
opção escolhida pelo utilizador for a opção da estação 3, salta para a etiqueta EscolhaBilhetes	
MOV R5, BEstacao4Preco	; R5 tem o
preço da estação 4	
CMP R1, OpcaoEst4	; Compara
a opção escolhida pelo utilizador com a opção da estação 4	
JEQ EscolhaBilhetes	; Se a
opção escolhida pelo utilizador for a opção da estação 4, salta para a etiqueta EscolhaBilhetes	
MOV R5, BEstacao5Preco	; R5 tem o
preço da estação 5	
CMP R1, OpcaoEst5	; Compara
a opção escolhida pelo utilizador com a opção da estação 5	
JEQ EscolhaBilhetes	; Se a
opção escolhida pelo utilizador for a opção da estação 5, salta para a etiqueta EscolhaBilhetes	
CMP R1, OpcaoCancelarCompra	; Compara a opção
escolhida pelo utilizador com a opção de cancelar a compra	

JEQ VoltarInicioCompra ; Se a
opção escolhida pelo utilizador for a opção de cancelar a compra, salta para a etiqueta
VoltarInicioCompra

CALL RotinaErro ; Se
a opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
RotinaErro

JMP RotinaComprar ; Salta para
a etiqueta RotinaComprar novamente

VoltarInicioCompra: ; Etiqueta
que permite ao utilizador voltar ao menu inicial usando um JMP para a etiqueta Ligado pois
este tem alcance suficiente

JMP Ligado

EscolhaBilhetes:

MOV R2, Display_EscolheuBilhete ; R2 tem o
endereço do display da escolha de bilhetes

CALL MostraDisplay ; Mostra o
display da escolha de bilhetes

CALL LimpaPerifericos ; Limpa os
periféricos

MOV R6, PrecoTotalCompra ; R6 tem o
endereço do preço total da compra

MOV R7, [R6] ;
R7 tem o valor do preço total da compra

ADD R7, R5 ;
Adiciona o preço do bilhete escolhido ao preço total da compra

MOV [R6], R7 ;
Atualiza o preço total da compra

LeOpcaoEscolhaBilhetes:

MOV R0, PER_EN ;
R0 tem o endereço do periférico de entrada

MOVB R1, [R0] ;
R1 tem o valor do periférico de entrada

CMP R1, 0 ;
Compara o valor do periférico de entrada com 0

JEQ LeOpcaoEscolhaBilhetes ; Se o valor
do periférico de entrada for 0, repete o ciclo

CMP R1, OpcaoSimEscolherBilhete ; Compara a opção
 escolhida pelo utilizador com a opção de sim escolher o bilhete
 JEQ RotinaComprar ; Se a
 opção escolhida pelo utilizador for a opção de sim escolher o bilhete, salta para a etiqueta
 RotinaComprar

CMP R1, OpcaoNaoEscolherBilhete ; Compara a opção
 escolhida pelo utilizador com a opção de não escolher o bilhete
 JEQ MostraPrecoTotal ; Se a
 opção escolhida pelo utilizador for a opção de não escolher o bilhete, salta para a etiqueta
 MostraPrecoTotal

CALL RotinaErro ; Se a
 opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
 RotinaErro

JMP EscolhaBilhetes ; Salta para a
 etiqueta EscolhaBilhetes novamente

;Mostra Preço Total da
 Compra-----

MostraPrecoTotal:

MOV R2, Display_PrecioTotal ; R2 tem o
 endereço do display do preço total

CALL MostraDisplay ; Mostra o
 display do preço total

CALL LimpaPerifericos ; Limpa os
 periféricos

MOV R6, PrecioTotalCompra ; R6 tem o
 endereço do preço total da compra

MOV R8, ValorEmFaltaCompra ; R8 tem o
 endereço do valor em falta da compra

MOV R7, [R6] ;
 R7 tem o valor do preço total da compra

MOV [R8], R7 ;
 Atualiza o valor em falta da compra

CALL EscreverPreco ; Chama a
 rotina EscreverPreco, que escreve o preço total da compra no display

LeOpcaoMostraPrecoTotal:

MOV R0, PER_EN ;
 R0 tem o endereço do periférico de entrada

```

        MOVB R1, [R0] ;
R1 tem o valor do periferico de entrada
        CMP R1, 0 ;
Compara o valor do periferico de entrada com 0
        JEQ LeOpcaoMostraPrecoTotal ; Se o valor do
periferico de entrada for 0, repete o ciclo

        CMP R1, OpcaoContinuarPrecoTotal ; Compara a opção
escolhida pelo utilizador com a opção de continuar
        JEQ InserirDinheiro1 ; Se a opção
escolhida pelo utilizador for a opção de continuar, salta para a etiqueta InserirDinheiro1

        CMP R1, OpcaoCancelarPrecoTotal ; Compara a opção
escolhida pelo utilizador com a opção de cancelar
        JEQ VoltarInicioCompra ; Se a
opção escolhida pelo utilizador for a opção de cancelar, salta para a etiqueta
VoltarInicioCompra

        CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
RotinaErro
        JMP MostraPrecoTotal ; Salta para
a etiqueta MostraPrecoTotal novamente

;Inserir
Dinheiro-----
-----
InserirDinheiro1:

        MOV R2, Display_InserirDinheiro1 ; R2 tem o endereço do
display da inserção de dinheiro
        CALL MostraDisplay ; Mostra o
display da inserção de dinheiro
        CALL LimpaPerifericos ; Limpa os
periféricos

LeOpcaoInserirDinheiro1:

        MOV R0, PER_EN ;
R0 tem o endereço do periferico de entrada
        MOVB R1, [R0] ;
R1 tem o valor do periferico de entrada
        CMP R1, 0 ;
Compara o valor do periferico de entrada com 0

```

JEQ LeOpcaoInserirDinheiro1 ; Se o valor do
periférico de entrada for 0, repete o ciclo

MOV R5, Valor10Centimos ; R5 tem o valor
de 10 centimos

CMP R1, Opcao10CentimosInserir1 ; Compara a opção
escolhida pelo utilizador com a opção de 10 centimos

JEQ InseriuDinheiro ; Se a opção
escolhida pelo utilizador for a opção de 10 centimos, salta para a etiqueta InseriuDinheiro

MOV R5, Valor20Centimos ; R5 tem o valor
de 20 centimos

CMP R1, Opcao20CentimosInserir1 ; Compara a opção
escolhida pelo utilizador com a opção de 20 centimos

JEQ InseriuDinheiro ; Se a opção
escolhida pelo utilizador for a opção de 20 centimos, salta para a etiqueta InseriuDinheiro

MOV R5, Valor50Centimos ; R5 tem o valor
de 50 centimos

CMP R1, Opcao50CentimosInserir1 ; Compara a opção
escolhida pelo utilizador com a opção de 50 centimos

JEQ InseriuDinheiro ; Se a opção
escolhida pelo utilizador for a opção de 50 centimos, salta para a etiqueta InseriuDinheiro

CMP R1, OpcaoSeguinteInserir1 ; Compara a opção
escolhida pelo utilizador com a opção de seguinte

JEQ InserirDinheiro2 ; Se a opção
escolhida pelo utilizador for a opção de seguinte, salta para a etiqueta InserirDinheiro2

CMP R1, OpcaoCancelarInserir1 ; Compara a opção
escolhida pelo utilizador com a opção de cancelar

JEQ VoltarInicioCompra ; Se a
opção escolhida pelo utilizador for a opção de cancelar, salta para a etiqueta
VoltarInicioCompra

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
RotinaErro

JMP InserirDinheiro1 ; Salta para a
etiqueta InserirDinheiro1 novamente

InserirDinheiro2:

MOV R2, Display_InserirDinheiro2 ; R2 tem o endereço do
display da inserção de dinheiro
CALL MostraDisplay ; Mostra o
display da inserção de dinheiro
CALL LimpaPerifericos ; Limpa os
periféricos

LeOpcaoInserirDinheiro2:

MOV R0, PER_EN ;
R0 tem o endereço do periférico de entrada
MOVB R1, [R0] ;
R1 tem o valor do periférico de entrada
CMP R1, 0 ;
Compara o valor do periférico de entrada com 0
JEQ LeOpcaoInserirDinheiro2 ; Se o valor do
periférico de entrada for 0, repete o ciclo

MOV R5, Valor1Euro ;
R5 tem o valor de 1 euro
CMP R1, Opcao1EuroInserir2 ; Compara
a opção escolhida pelo utilizador com a opção de 1 euro
JEQ InserirDinheiro ; Se a opção
escolhida pelo utilizador for a opção de 1 euro, salta para a etiqueta InserirDinheiro

MOV R5, Valor2Euros ; R5 tem o
valor de 2 euros
CMP R1, Opcao2EuroInserir2 ; Compara
a opção escolhida pelo utilizador com a opção de 2 euros
JEQ InserirDinheiro ; Se a opção
escolhida pelo utilizador for a opção de 2 euros, salta para a etiqueta InserirDinheiro

MOV R5, Valor5Euros ; R5 tem o
valor de 5 euros
CMP R1, Opcao5EurosInserir2 ; Compara a opção
escolhida pelo utilizador com a opção de 5 euros
JEQ InserirDinheiro ; Se a opção
escolhida pelo utilizador for a opção de 5 euros, salta para a etiqueta InserirDinheiro

CMP R1, OpcaoSeguinteInserir2 ; Compara a opção
escolhida pelo utilizador com a opção de seguinte
JEQ InserirDinheiro3 ; Se a opção
escolhida pelo utilizador for a opção de seguinte, salta para a etiqueta InserirDinheiro3

CMP R1, OpcaoCancelarInserir2 ; Compara a opção
escolhida pelo utilizador com a opção de cancelar

JEQ VoltarInicioCompra2 ; Se a opção
escolhida pelo utilizador for a opção de cancelar, salta para a etiqueta VoltarInicioCompra2

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
RotinaErro

JMP InserirDinheiro2 ; Salta para a
etiqueta InserirDinheiro2 novamente

InserirDinheiro3:

MOV R2, Display_InserirDinheiro3 ; R2 tem o endereço do
display da inserção de dinheiro

CALL MostraDisplay ; Mostra o
display da inserção de dinheiro

CALL LimpaPerifericos ; Limpa os
periféricos

LeOpcaoInserirDinheiro3:

MOV R0, PER_EN ;
R0 tem o endereço do periférico de entrada

MOVB R1, [R0] ;
R1 tem o valor do periférico de entrada

CMP R1, 0 ;
Compara o valor do periférico de entrada com 0

JEQ LeOpcaoInserirDinheiro3 ; Se o valor do
periférico de entrada for 0, repete o ciclo

MOV R5, Valor10Euros ; R5 tem o
valor de 10 euros

CMP R1, Opcao10EuroInserir3 ; Compara a opção
escolhida pelo utilizador com a opção de 10 euros

JEQ InseriuDinheiro ; Se a opção
escolhida pelo utilizador for a opção de 10 euros, salta para a etiqueta InseriuDinheiro

MOV R5, Valor20Euros ; R5 tem o
valor de 20 euros

CMP R1, Opcao20EuroInserir3 ; Compara a opção
escolhida pelo utilizador com a opção de 20 euros

JEQ InseriuDinheiro ; Se a opção
escolhida pelo utilizador for a opção de 20 euros, salta para a etiqueta InseriuDinheiro

CMP R1, OpcaoSeguinteInserir3 ; Compara a opção
escolhida pelo utilizador com a opção de seguinte

JEQ InserirDinheiro1 ; Se a opção
escolhida pelo utilizador for a opção de seguinte, salta para a etiqueta InserirDinheiro1

CMP R1, OpcaoCancelarInserir3 ; Compara a opção
escolhida pelo utilizador com a opção de cancelar

JEQ VoltarInicioCompra2 ; Se a opção
escolhida pelo utilizador for a opção de cancelar, salta para a etiqueta VoltarInicioCompra2

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
RotinaErro

JMP InserirDinheiro3 ; Salta para a
etiqueta InserirDinheiro3 novamente

InseriuDinheiro:

MOV R2, Display_InseriuDinheiro ; R2 tem o
endereço do display da inserção de dinheiro

CALL MostraDisplay ; Mostra o
display da inserção de dinheiro

CALL LimpaPerifericos ; Limpa os
periféricos

CALL AdicionaStock ; Chama a
rotina AdicionaStock, que adiciona o valor do dinheiro inserido ao stock da máquina

MOV R10, 0 ;
R10 tem o valor 0 para sinalizar que o preço em falta não foi pago

CALL AtualizarPrecoEmFalta ; Chama a
rotina AtualizarPrecoEmFalta, que atualiza o preço em falta da compra e o valor total
inserido pelo utilizador

CMP R10, 1 ;
Compara o valor de R10 com 1, que quer dizer que o preço em falta foi pago

JEQ Pago ; Se
o valor de R10 for 1, salta para a etiqueta Pago

LeOpcaoInseriuDinheiro:

```

MOV R0, PER_EN ;
R0 tem o endereço do periférico de entrada
MOVB R1, [R0] ;
R1 tem o valor do periférico de entrada
CMP R1, 0 ;
Compara o valor do periférico de entrada com 0
JEQ LeOpcaoInseriuDinheiro ; Se o valor
do periférico de entrada for 0, repete o ciclo

```

```

CMP R1, OpcaoContinuarInseriuDinheiro ; Compara a opção
escolhida pelo utilizador com a opção de continuar
JEQ InserirDinheiro1 ; Se a opção
escolhida pelo utilizador for a opção de continuar, salta para a etiqueta InserirDinheiro1

```

```

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for a opção de continuar, chama a rotina RotinaErro
JMP InseriuDinheiro ; Salta para a
etiqueta InseriuDinheiro novamente

```

```

;Talão-----
-----

```

Pago:

```

MOV R2, Display_Talao ; R2 tem o
endereço do display do talão
CALL MostraDisplay ; Mostra o
display do talão
CALL LimpaPerifericos ; Limpa os
periféricos

```

```

MOV R5, PrecoTotalCompra ; R5 tem o
endereço do preço total da compra
MOV R6, [R5] ;
R6 tem o valor do preço total da compra
MOV R5, ValorInseridoCompra ; R5 tem o
endereço do valor total inserido pelo utilizador
MOV R7, [R5] ;
R7 tem o valor do valor total inserido pelo utilizador
MOV R5, TrocoCompra ;
R5 tem o endereço do preço em falta da compra
SUB R7, R6 ;
Subtrai o preço total da compra ao valor total inserido pelo utilizador
MOV [R5], R7 ;
Atualiza o troco da compra

```

```

MOV R10, 0 ;
R10 tem o valor 0 para saber se tem espaço para criar um novo PEPE

CALL CriarPEPE ;
Chama a rotina CriarPEPE, que cria um novo PEPE

CMP R10, 1 ;
Compara o valor de R10 com 1, que quer dizer que não tem espaço para criar um novo PEPE
JEQ SemEspaco ; Se
o valor de R10 for 1, salta para a etiqueta SemEspaco

CALL EscreverTalao ; Chama a
rotina EscreverTalao, que escreve os valores do número do PEPE criado, preco total, valor
total inserido e troco no talão

LeOpcaoPago:

MOV R0, PER_EN ;
R0 tem o endereco do periferico de entrada
MOVB R1, [R0] ;
R1 tem o valor do periferico de entrada
CMP R1, 0 ;
Compara o valor do periferico de entrada com 0
JEQ LeOpcaoPago ; Se o valor
do periferico de entrada for 0, repete o ciclo

CMP R1, OpcaoContinuarTalao ; Compara a opção
escolhida pelo utilizador com a opção de continuar
JEQ PrimeiraCompra ; Se a
opção escolhida pelo utilizador for a opção de continuar, salta para a etiqueta
PrimeiraCompra

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for a opção de continuar, chama a rotina RotinaErro
JMP Pago ;
Salta para a etiqueta Pago novamente

VoltarInicioCompra2: ; Etiqueta
que permite ao utilizador voltar ao menu inicial usando um JMP para a etiqueta Ligado pois
este tem alcance suficiente
JMP Ligado

```

;Erro de espaço para

PEPE-----

SemEspaco:

MOV R2, Display_ErroEspacoInsuficiente ; R2 tem o endereço do
display de erro de espaço insuficiente
CALL MostraDisplay ; Mostra o
display de erro de espaço insuficiente
CALL LimpaPerifericos ; Limpa os
periféricos

LeOpcaoSemEspaco:

MOV R0, PER_EN ;
R0 tem o endereço do periférico de entrada
MOVB R1, [R0] ;
R1 tem o valor do periférico de entrada
CMP R1, 0 ;
Compara o valor do periférico de entrada com 0
JEQ LeOpcaoSemEspaco ; Se o valor
do periférico de entrada for 0, repete o ciclo

CMP R1, OpcaoVoltarErro ; Compara a opção
escolhida pelo utilizador com a opção de voltar
JEQ VoltarInicioCompra2 ; Se a opção
escolhida pelo utilizador for a opção de voltar, salta para a etiqueta VoltarInicioCompra2

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for a opção de voltar, chama a rotina RotinaErro
JMP SemEspaco ;
Salta para a etiqueta SemEspaco novamente

;Promoção da Primeira

Compra-----

PrimeiraCompra:

MOV R2, Display_PrimeiraCompra ; R2 tem o
endereço do display da primeira compra
CALL MostraDisplay ; Mostra o
display da primeira compra
CALL LimpaPerifericos ; Limpa os
periféricos

```

MOV R5, PrecoTotalCompra ; R5 tem o
endereço do preço total da compra
MOV R10, [R5] ;
R10 tem o valor do preço total da compra
CALL AtualizarSaldoPEPE ; Chama a
rotina AtualizarSaldoPEPE, que atualiza o saldo do PEPE atual

LeOpcaoPrimeiraCompra:

MOV R0, PER_EN ;
R0 tem o endereço do periférico de entrada
MOVB R1, [R0] ;
R1 tem o valor do periférico de entrada
CMP R1, 0 ;
Compara o valor do periférico de entrada com 0
JEQ LeOpcaoPrimeiraCompra ; Se o valor
do periférico de entrada for 0, repete o ciclo

CMP R1, OpcaoContinuarPrimeiraCompra ; Compara a opção
escolhida pelo utilizador com a opção de continuar
JEQ Troco ; Se
a opção escolhida pelo utilizador for a opção de continuar, salta para a etiqueta Troco

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for a opção de continuar, chama a rotina RotinaErro
JMP PrimeiraCompra ; Salta para
a etiqueta PrimeiraCompra novamente

;Troco-----
-----
Troco:

MOV R5, TrocoCompra ;
R5 tem o endereço do troco da compra
MOV R6, [R5] ;
R6 tem o valor do troco da compra

CMP R6, 0 ;
Compara o valor do troco da compra com 0
JGT DevolverTroco ; Se o valor
do troco da compra for maior que 0, salta para a etiqueta DevolverTroco
JMP Ligado ; Se
o valor do troco da compra for igual a 0, volta ao menu principal

```

DevolverTroco:

CALL DevolveTroco ; Chama a
rotina DevolveTroco, que retira da máquina os valores monetários correspondentes ao troco
da compra

MOV R10, 0 ;
R10 tem o valor 0 para fins de verificação de devolução de troco

CALL RetiraStock ; Chama a
rotina RetiraStock, que retira da máquina os valores monetários correspondentes ao troco da
compra

CMP R10, 1 ;
Compara o valor de R10 com 1, que quer dizer que não há como devolver o troco
JEQ SemTroco ; Se
o valor de R10 for 1, salta para a etiqueta SemTroco
JMP DevolverTroco1 ; Se o valor
de R10 for 0, salta para a etiqueta DevolverTroco1

;Erro de falta de troco na
máquina-----

--

SemTroco:

MOV R2, Display_SemTroco ; R2 tem o
endereço do display de sem troco
CALL MostraDisplay ; Mostra o
display de sem troco
CALL LimpaPerifericos ; Limpa os
periféricos

LeOpcaoSemTroco:

MOV R0, PER_EN ;
R0 tem o endereço do periférico de entrada
MOVB R1, [R0] ;
R1 tem o valor do periférico de entrada
CMP R1, 0 ;
Compara o valor do periférico de entrada com 0
JEQ LeOpcaoSemTroco ; Se o valor
do periférico de entrada for 0, repete o ciclo

CMP R1, OpcaoContinuarSemTroco ; Compara
a opção escolhida pelo utilizador com a opção de continuar
JEQ ObrigadoCompra ; Se
a opção escolhida pelo utilizador for a opção de continuar, salta para a etiqueta
ObrigadoCompra

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for a opção de continuar, chama a rotina RotinaErro
JMP SemTroco ;
Salta para a etiqueta SemTroco novamente

DevolverTroco1:

MOV R2, Display_TrocoDevolvido1 ; R2 tem o
endereço do display da devolução do troco
CALL MostraDisplay ; Mostra o
display da devolução do troco
CALL LimpaPerifericos ; Limpa os
periféricos

MOV R6, 3 ;
R6 tem o valor 3, que é a quantidade de valores a escrever
MOV R7, TrocoMoedas10 ; R7 tem o
endereço do primeiro valor a escrever
CALL EscreverTroco ; Chama a
rotina EscreverTroco, que escreve os valores monetários devolvidos no display

LeOpcaoDevolverTroco1:

MOV R0, PER_EN ;
R0 tem o endereço do periférico de entrada
MOVB R1, [R0] ;
R1 tem o valor do periférico de entrada
CMP R1, 0 ;
Compara o valor do periférico de entrada com 0
JEQ LeOpcaoDevolverTroco1 ; Se o valor
do periférico de entrada for 0, repete o ciclo

CMP R1, OpcaoSeguinteTrocoDevolvido ; Compara a opção
escolhida pelo utilizador com a opção de seguinte
JEQ DevolverTroco2 ; Se a
opção escolhida pelo utilizador for a opção de seguinte, salta para a etiqueta DevolverTroco2

CMP R1, OpcaoSairTrocoDevolvido ; Compara a opção
 escolhida pelo utilizador com a opção de sair
 JEQ ObrigadoCompra ; Se
 a opção escolhida pelo utilizador for a opção de sair, salta para a etiqueta ObrigadoCompra

CALL RotinaErro ; Se a
 opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
 RotinaErro
 JMP DevolverTroco1 ; Salta para
 a etiqueta DevolverTroco1 novamente

DevolverTroco2:

MOV R2, Display_TrocoDevolvido2 ; R2 tem o
 endereço do display da devolução do troco
 CALL MostraDisplay ; Mostra o
 display da devolução do troco
 CALL LimpaPerifericos ; Limpa os
 periféricos

MOV R6, 3 ;
 R6 tem o valor 3, que é a quantidade de valores a escrever
 MOV R7, TrocoMoedas1 ; R7 tem o
 endereço do primeiro valor a escrever
 CALL EscreverTroco ; Chama a
 rotina EscreverTroco, que escreve os valores monetários devolvidos no display

LeOpcaoDevolverTroco2:

MOV R0, PER_EN ;
 R0 tem o endereço do periferico de entrada
 MOVB R1, [R0] ;
 R1 tem o valor do periferico de entrada
 CMP R1, 0 ;
 Compara o valor do periferico de entrada com 0
 JEQ LeOpcaoDevolverTroco2 ; Se o valor
 do periferico de entrada for 0, repete o ciclo

CMP R1, OpcaoSeguinteTrocoDevolvido ; Compara a opção
 escolhida pelo utilizador com a opção de seguinte
 JEQ DevolverTroco3 ; Se a
 opção escolhida pelo utilizador for a opção de seguinte, salta para a etiqueta DevolverTroco3

CMP R1, OpcaoSairTrocoDevolvido ; Compara a opção
escolhida pelo utilizador com a opção de sair

JEQ ObrigadoCompra ; Se
a opção escolhida pelo utilizador for a opção de sair, salta para a etiqueta ObrigadoCompra

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
RotinaErro

JMP DevolverTroco2 ; Salta para
a etiqueta DevolverTroco2 novamente

DevolverTroco3:

MOV R2, Display_TrocoDevolvido3 ; R2 tem o
endereço do display da devolução do troco

CALL MostraDisplay ; Mostra o
display da devolução do troco

CALL LimpaPerifericos ; Limpa os
periféricos

MOV R6, 2 ;
R6 tem o valor 2, que é a quantidade de valores a escrever

MOV R7, TrocoNotas10 ; R7 tem o
endereço do primeiro valor a escrever

CALL EscreverTroco ; Chama a
rotina EscreverTroco, que escreve os valores monetários devolvidos no display

LeOpcaoDevolverTroco3:

MOV R0, PER_EN ;
R0 tem o endereço do periferico de entrada

MOVB R1, [R0] ;
R1 tem o valor do periferico de entrada

CMP R1, 0 ;
Compara o valor do periferico de entrada com 0

JEQ LeOpcaoDevolverTroco3 ; Se o valor
do periferico de entrada for 0, repete o ciclo

CMP R1, OpcaoSeguinteTrocoDevolvido ; Compara a opção
escolhida pelo utilizador com a opção de seguinte

JEQ DevolverTroco1 ; Se a
opção escolhida pelo utilizador for a opção de seguinte, salta para a etiqueta DevolverTroco1

CMP R1, OpcaoSairTrocoDevolvido ; Compara a opção
escolhida pelo utilizador com a opção de sair

JEQ ObrigadoCompra ; Se
a opção escolhida pelo utilizador for a opção de sair, salta para a etiqueta ObrigadoCompra

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
RotinaErro

JMP DevolverTroco3 ; Salta para
a etiqueta DevolverTroco3 novamente

;Agradecimento pela
Compra-----

ObrigadoCompra:

MOV R2, Display_ObrigadoCompra ; R2 tem o
endereço do display de agradecimento pela compra

CALL MostraDisplay ; Mostra o
display de agradecimento pela compra

CALL LimpaPerifericos ; Limpa os
periféricos

LeOpcaoObrigadoCompra:

MOV R0, PER_EN ;
R0 tem o endereço do periférico de entrada

MOVB R1, [R0] ;
R1 tem o valor do periférico de entrada

CMP R1, 0 ;
Compara o valor do periférico de entrada com 0

JEQ LeOpcaoObrigadoCompra ; Se o valor
do periférico de entrada for 0, repete o ciclo

CMP R1, OpcaoContinuarObrigadoCompra ; Compara a opção
escolhida pelo utilizador com a opção de continuar

JEQ VoltarInicioCompra2 ; Se a opção
escolhida pelo utilizador for a opção de continuar, salta para a etiqueta VoltarInicioCompra2

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for a opção de continuar, chama a rotina RotinaErro

JMP ObrigadoCompra ;
Salta para a etiqueta ObrigadoCompra novamente

```

;-----
;
; Etiqueta
Usar Cartão
;-----
; Etiqueta que permite ao utilizador usar um cartão PEPE para comprar bilhetes ou recarregar
o saldo do cartão
;-----

;Verificação do
Número-----

RotinaUsarCartao:

    MOV R2, Display_CodigoPEPE                ; R2 tem o
    endereço do display do código do PEPE
    CALL MostraDisplay                        ; Mostra o
    display do código do PEPE
    CALL LimpaPerifericos                    ; Limpa os
    periféricos

LeOpcaoUsarCartao:

    MOV R0, PER_EN                            ;
    R0 tem o endereço do periférico de entrada
    MOVB R1, [R0]                            ; R1 tem o
    valor do periférico de entrada
    CMP R1, 0                                ; Compara
    o valor do periférico de entrada com 0
    JEQ LeOpcaoUsarCartao                    ; Se o valor
    do periférico de entrada for 0, repete o ciclo

    CMP R1, OpcaoContinuarCartao              ; Compara a opção
    escolhida pelo utilizador com a opção de continuar
    JEQ VerificarNPEPE                        ; Se a
    opção escolhida pelo utilizador for a opção de continuar, salta para a etiqueta
    VerificarNPEPE

    CMP R1, OpcaoCancelarCartao               ; Compara a opção
    escolhida pelo utilizador com a opção de cancelar

```

JEQ VoltarInicioUsarCartao ; Se a opção
escolhida pelo utilizador for a opção de cancelar, salta para a etiqueta VoltarInicioUsarCartao

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
RotinaErro

JMP RotinaUsarCartao ; Salta para
a etiqueta RotinaUsarCartao novamente

VoltarInicioUsarCartao: ; Etiqueta que
permite ao utilizador voltar ao menu inicial usando um JMP para a etiqueta Ligado pois este
tem alcance suficiente

JMP Ligado

VerificarNPEPE:

MOV R7, 0 ;
R7 tem o valor 0 para fins de verificação do número do PEPE introduzido pelo utilizador

CALL VerificaPEPE ; Chama a
rotina VerificaPEPE, que verifica se o PEPE introduzido pelo utilizador existe na base de
dados dos PEPEs

CMP R7, 1 ;
Compara o valor de R7 com 1, que quer dizer que o PEPE introduzido pelo utilizador existe
JEQ ConsultarCartao ; Se o valor
de R7 for 1, salta para a etiqueta ConsultarCartao

;PEPE Não

Encontrado-----

NPEPEIntroduzidoErrado:

MOV R2, Display_NPEPEErrado ; R2 tem o
endereço do display de número do PEPE errado

CALL MostraDisplay ; Mostra o
display de número do PEPE errado

CALL LimpaPerifericos ; Limpa os
periféricos

LeOpcaoNPEPEIntroduzidoErrado:

MOV R0, PER_EN ;
R0 tem o endereço do periférico de entrada

MOVB R1, [R0] ;
R1 tem o valor do periférico de entrada

```

        CMP R1, 0
        Compara o valor do periferico de entrada com 0
        JEQ LeOpcaoNPEPEIntroduzidoErrado ; Se o valor do
        periferico de entrada for 0, repete o ciclo

        CMP R1, OpcaoVoltarNPEPEErrado ; Compara
        a opção escolhida pelo utilizador com a opção de voltar
        JEQ RotinaUsarCartao ; Se a
        opção escolhida pelo utilizador for a opção de voltar, salta para a etiqueta RotinaUsarCartao

        CALL RotinaErro ; Se a
        opção escolhida pelo utilizador não for a opção de voltar, chama a rotina RotinaErro
        JMP NPEPEIntroduzidoErrado ; Salta para
        a etiqueta NPEPEIntroduzidoErrado novamente

;Consultar/Gerir Cartão
PEPE-----
-----
ConsultarCartao:

        MOV R2, Display_MenuUsarPEPE ; R2 tem o
        endereço do display do menu de usar PEPE
        CALL MostraDisplay ; Mostra o
        display do menu de usar PEPE
        CALL LimpaPerifericos ; Limpa os
        periféricos

        CALL ReiniciarVarCompra ; Chama a
        rotina ReiniciarVarCompra, que reinicia as variáveis da compra

        CALL EscreverCartao ;
        Chama a rotina EscreverCartao, que escreve o número do PEPE atual e o saldo desse

LeOpcaoConsultarCartao:

        MOV R0, PER_EN ;
        R0 tem o endereço do periferico de entrada
        MOVB R1, [R0] ;
        R1 tem o valor do periferico de entrada
        CMP R1, 0 ;
        Compara o valor do periferico de entrada com 0
        JEQ LeOpcaoConsultarCartao ; Se o valor
        do periferico de entrada for 0, repete o ciclo

```

CMP R1, OpcaoComprarPEPE ; Compara
a opção escolhida pelo utilizador com a opção de comprar
JEQ ComprarPEPE ; Se a
opção escolhida pelo utilizador for a opção de comprar, salta para a etiqueta ComprarPEPE

CMP R1, OpcaoRecarregarPEPE ; Compara a opção
escolhida pelo utilizador com a opção de recarregar
JEQ InserirDinheiroPEPEIntermedio ; Se a opção
escolhida pelo utilizador for a opção de recarregar, salta para a etiqueta
InserirDinheiroPEPEIntermedio

CMP R1, OpcaoVoltarPEPE ; Compara a opção
escolhida pelo utilizador com a opção de voltar
JEQ VoltarInicioUsarCartao ; Se a opção
escolhida pelo utilizador for a opção de voltar, salta para a etiqueta VoltarInicioUsarCartao

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
RotinaErro

JMP ConsultarCartao ; Salta para a
etiqueta ConsultarCartao novamente

InserirDinheiroPEPEIntermedio: ; Etiqueta que
permite ao utilizador ir para o menu de recarregar o saldo do cartão PEPE usando um JMP
para a etiqueta InserirDinheiroPEPE1 pois este tem alcance suficiente

JMP InserirDinheiroPEPE1

;Compra com

PEPE-----

ComprarPEPE:

MOV R2, Display_MenuComprar ; R2 tem o
endereço do display do menu de comprar

CALL MostraDisplay ;
Mostra o display do menu de comprar

CALL LimpaPerifericos ; Limpa os
periféricos

CALL ReiniciarVarCompra ; Chama a rotina
ReiniciarVarCompra, que reinicia as variáveis da compra

LeOpcaoComprarPEPE:

MOV R0, PER_EN	;
R0 tem o endereço do periférico de entrada	
MOVB R1, [R0]	;
R1 tem o valor do periférico de entrada	
CMP R1, 0	;
Compara o valor do periférico de entrada com 0	
JEQ LeOpcaoComprarPEPE	; Se o valor
do periférico de entrada for 0, repete o ciclo	
MOV R5, BEstacao1Preco	; R5 tem o
preço da estação 1	
CMP R1, OpcaoEst1	; Compara
a opção escolhida pelo utilizador com a opção da estação 1	
JEQ EscolhaBilhetesPEPE	; Se a opção
escolhida pelo utilizador for a opção da estação 1, salta para a etiqueta EscolhaBilhetesPEPE	
MOV R5, BEstacao2Preco	; R5 tem o
preço da estação 2	
CMP R1, OpcaoEst2	; Compara
a opção escolhida pelo utilizador com a opção da estação 2	
JEQ EscolhaBilhetesPEPE	; Se a opção
escolhida pelo utilizador for a opção da estação 2, salta para a etiqueta EscolhaBilhetesPEPE	
MOV R5, BEstacao3Preco	; R5 tem o
preço da estação 3	
CMP R1, OpcaoEst3	; Compara
a opção escolhida pelo utilizador com a opção da estação 3	
JEQ EscolhaBilhetesPEPE	; Se a opção
escolhida pelo utilizador for a opção da estação 3, salta para a etiqueta EscolhaBilhetesPEPE	
MOV R5, BEstacao4Preco	; R5 tem o
preço da estação 4	
CMP R1, OpcaoEst4	; Compara
a opção escolhida pelo utilizador com a opção da estação 4	
JEQ EscolhaBilhetesPEPE	; Se a opção
escolhida pelo utilizador for a opção da estação 4, salta para a etiqueta EscolhaBilhetesPEPE	
MOV R5, BEstacao5Preco	; R5 tem o
preço da estação 5	
CMP R1, OpcaoEst5	; Compara
a opção escolhida pelo utilizador com a opção da estação 5	
JEQ EscolhaBilhetesPEPE	; Se a opção
escolhida pelo utilizador for a opção da estação 5, salta para a etiqueta EscolhaBilhetesPEPE	

CMP R1, OpcaoCancelarCompra ; Compara a opção
escolhida pelo utilizador com a opção de cancelar

JEQ ConsultarCartao ; Se a opção
escolhida pelo utilizador for a opção de cancelar, salta para a etiqueta ConsultarCartao

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
RotinaErro

JMP ComprarPEPE ; Salta para
a etiqueta ComprarPEPE novamente

EscolhaBilhetesPEPE:

MOV R2, Display_EscolheuBilhete ; R2 tem o endereço do
display de escolha de bilhete

CALL MostraDisplay ;
Mostra o display de escolha de bilhete

CALL LimpaPerifericos ; Limpa os
periféricos

MOV R6, PrecoTotalCompra ; R6 tem o
endereço do preço total da compra

MOV R7, [R6] ;
R7 tem o valor do preço total da compra

ADD R7, R5 ;
Adiciona o preço do bilhete escolhido ao preço total da compra

MOV [R6], R7 ;
Atualiza o preço total da compra

LeOpcaoEscolhaBilhetesPEPE:

MOV R0, PER_EN ;
R0 tem o endereço do periférico de entrada

MOVB R1, [R0] ;
R1 tem o valor do periférico de entrada

CMP R1, 0 ;
Compara o valor do periférico de entrada com 0

JEQ LeOpcaoEscolhaBilhetesPEPE ; Se o valor do
periférico de entrada for 0, repete o ciclo

CMP R1, OpcaoSimEscolherBilhete ; Compara a opção
escolhida pelo utilizador com a opção de sim

JEQ EscolhaBilhetesPEPE ; Se a opção
escolhida pelo utilizador for a opção de sim, salta para a etiqueta EscolhaBilhetesPEPE

CMP R1, OpcaoNaoEscolherBilhete ; Compara a opção
 escolhida pelo utilizador com a opção de não
 JEQ MostraPrecoTotalPEPE ; Se a opção
 escolhida pelo utilizador for a opção de não, salta para a etiqueta MostraPrecoTotalPEPE

CALL RotinaErro ; Se a
 opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
 RotinaErro

JMP EscolhaBilhetesPEPE ; Salta para a
 etiqueta EscolhaBilhetesPEPE novamente

MostraPrecoTotalPEPE:

MOV R2, Display_PrecioTotal ; R2 tem o
 endereço do display do preço total

CALL MostraDisplay ;
 Mostra o display do preço total

CALL LimpaPerifericos ; Limpa os
 periféricos

MOV R6, PrecioTotalCompra ; R6 tem o
 endereço do preço total da compra

CALL EscreverPrecio ; Chama a
 rotina EscreverPrecio, que escreve o preço total da compra no display

LeOpcaoMostraPrecoTotalPEPE:

MOV R0, PER_EN ;
 R0 tem o endereço do periférico de entrada

MOVB R1, [R0] ;
 R1 tem o valor do periférico de entrada

CMP R1, 0 ;
 Compara o valor do periférico de entrada com 0

JEQ LeOpcaoMostraPrecoTotalPEPE ; Se o valor do
 periférico de entrada for 0, repete o ciclo

CMP R1, OpcaoContinuarPrecioTotal ; Compara a opção
 escolhida pelo utilizador com a opção de continuar

JEQ VerificarSaldoPEPE ; Se a
 opção escolhida pelo utilizador for a opção de continuar, salta para a etiqueta
 VerificarSaldoPEPE

CMP R1, OpcaoCancelarPrecioTotal ; Compara a opção
 escolhida pelo utilizador com a opção de cancelar

JEQ ConsultarCartao ; Se a opção
escolhida pelo utilizador for a opção de cancelar, salta para a etiqueta ConsultarCartao

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
RotinaErro

JMP MostraPrecoTotalPEPE ; Salta para a
etiqueta MostraPrecoTotalPEPE novamente

VerificarSaldoPEPE:

MOV R6, PrecoTotalCompra ; R6 tem o
endereço do preço total da compra

MOV R7, [R6] ;
R7 tem o valor do preço total da compra

MOV R5, SaldoPEPEAtual ; R5 tem o
endereço do saldo do PEPE atual

MOV R6, [R5] ;
R6 tem o valor do saldo do PEPE atual

CMP R6, R7 ;
Compara o valor do saldo do PEPE atual com o preço total da compra

JLT SaldoInsuficientePEPE ; Se o valor do
saldo do PEPE atual for menor que o preço total da compra, salta para a etiqueta
SaldoInsuficientePEPE

JMP PagoPEPE ; Se
o valor do saldo do PEPE atual for maior ou igual ao preço total da compra, salta para a
etiqueta PagoPEPE

SaldoInsuficientePEPE:

MOV R2, Display_SaldoInsuficientePEPE ; R2 tem o endereço do
display de saldo insuficiente

CALL MostraDisplay ;
Mostra o display de saldo insuficiente

CALL LimpaPerifericos ; Limpa os
periféricos

LeOpcaoSaldoInsuficientePEPE:

MOV R0, PER_EN ;
R0 tem o endereço do periférico de entrada

MOVB R1, [R0] ;
R1 tem o valor do periférico de entrada

CMP R1, 0 ;
 Compara o valor do periférico de entrada com 0
 JEQ LeOpcaoSaldoInsuficientePEPE ; Se o valor do
 periférico de entrada for 0, repete o ciclo

CMP R1, OpcaoVoltarSaldoInsuficientePEPE ; Compara a opção
 escolhida pelo utilizador com a opção de voltar
 JEQ ConsultarCartao ; Se a opção
 escolhida pelo utilizador for a opção de voltar, salta para a etiqueta ConsultarCartao

CALL RotinaErro ; Se a
 opção escolhida pelo utilizador não for a opção de voltar, chama a rotina RotinaErro
 JMP SaldoInsuficientePEPE ; Salta para a
 etiqueta SaldoInsuficientePEPE novamente

PagoPEPE:

MOV R2, Display_TalaoPEPE ; R2 tem o
 endereço do display do talão do PEPE
 CALL MostraDisplay ;
 Mostra o display do talão do PEPE
 CALL LimpaPerifericos ; Limpa os
 periféricos

MOV R5, PrecoTotalCompra ; R5 tem o
 endereço do preço total da compra
 MOV R6, [R5] ;
 R6 tem o valor do preço total da compra
 MOV R5, SaldoPEPEAtual ; R5 tem o
 endereço do saldo do PEPE atual
 MOV R7, [R5] ;
 R7 tem o valor do saldo do PEPE atual
 SUB R7, R6 ;
 Subtrai o preço total da compra ao saldo do PEPE atual
 MOV [R5], R7 ;
 Atualiza o saldo do PEPE atual

MOV R10, SaldoPEPEAtual ; R10 tem o
 endereço do saldo do PEPE atual
 MOV R5, NPEPEAtual ;
 R5 tem o endereço do número do PEPE atual
 MOV R11, [R5] ;
 R11 tem o valor do número do PEPE atual

CALL AtualizarSaldoPEPE ; Chama a rotina
 AtualizarSaldoPEPE, que atualiza o saldo do PEPE atual
 CALL EscreverTalaoPEPE ; Chama a
 rotina EscreverTalaoPEPE, que escreve o talão do PEPE

LeOpcaoPagoPEPE:

MOV R0, PER_EN ;
 R0 tem o endereço do periférico de entrada
 MOVB R1, [R0] ;
 R1 tem o valor do periférico de entrada
 CMP R1, 0 ;
 Compara o valor do periférico de entrada com 0
 JEQ LeOpcaoPagoPEPE ; Se o valor
 do periférico de entrada for 0, repete o ciclo

CMP R1, OpcaoContinuarTalaoPEPE ; Compara a opção
 escolhida pelo utilizador com a opção de continuar
 JEQ ObrigadoCompraPEPE ; Se a
 opção escolhida pelo utilizador for a opção de continuar, salta para a etiqueta
 ObrigadoCompraPEPE

CALL RotinaErro ; Se a
 opção escolhida pelo utilizador não for a opção de continuar, chama a rotina RotinaErro
 JMP PagoPEPE ;
 Salta para a etiqueta PagoPEPE novamente

ObrigadoCompraPEPE:

MOV R2, Display_ObrigadoCompra ; R2 tem o
 endereço do display de agradecimento pela compra
 CALL MostraDisplay ; Mostra o
 display de agradecimento pela compra
 CALL LimpaPerifericos ; Limpa os
 periféricos

LeOpcaoObrigadoCompraPEPE:

MOV R0, PER_EN ;
 R0 tem o endereço do periférico de entrada
 MOVB R1, [R0] ;
 R1 tem o valor do periférico de entrada
 CMP R1, 0 ;
 Compara o valor do periférico de entrada com 0

JEQ LeOpcaoObrigadoCompraPEPE ; Se o valor
do periferico de entrada for 0, repete o ciclo

CMP R1, OpcaoContinuarObrigadoCompra ; Compara a opção
escolhida pelo utilizador com a opção de continuar

JEQ VoltarConsultarCartao ; Se a opção
escolhida pelo utilizador for a opção de continuar, salta para a etiqueta VoltarConsultarCartao

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for a opção de continuar, chama a rotina RotinaErro

JMP ObrigadoCompraPEPE ; Salta para
a etiqueta ObrigadoCompraPEPE novamente

VoltarConsultarCartao: ; Etiqueta
que permite ao utilizador voltar ao menu de usar cartão PEPE usando um JMP para a etiqueta
ConsultarCartao pois este tem alcance suficiente

JMP ConsultarCartao

;Recarregar

PEPE-----

InserirDinheiroPEPE1:

MOV R2, Display_InserirDinheiro1 ; R2 tem o endereço do
display de inserção de dinheiro

CALL MostraDisplay ;
Mostra o display de inserção de dinheiro

CALL LimpaPerifericos ; Limpa os
periféricos

LeOpcaoRecarregarPEPE1:

MOV R0, PER_EN ;
R0 tem o endereço do periferico de entrada

MOVB R1, [R0] ;
R1 tem o valor do periferico de entrada

CMP R1, 0 ;
Compara o valor do periferico de entrada com 0

JEQ LeOpcaoRecarregarPEPE1 ; Se o valor
do periferico de entrada for 0, repete o ciclo

MOV R5, Valor10Centimos ; R5 tem o valor
de 10 cêntimos

CMP R1, Opcao10CentimosInserir1 ; Compara a opção
escolhida pelo utilizador com a opção de 10 cêntimos

JEQ InseriuDinheiroPEPE ; Se a opção
escolhida pelo utilizador for a opção de 10 cêntimos, salta para a etiqueta
InseriuDinheiroPEPE

MOV R5, Valor20Centimos ; R5 tem o valor
de 20 cêntimos

CMP R1, Opcao20CentimosInserir1 ; Compara a opção
escolhida pelo utilizador com a opção de 20 cêntimos

JEQ InseriuDinheiroPEPE ; Se a opção
escolhida pelo utilizador for a opção de 20 cêntimos, salta para a etiqueta
InseriuDinheiroPEPE

MOV R5, Valor50Centimos ; R5 tem o valor
de 50 cêntimos

CMP R1, Opcao50CentimosInserir1 ; Compara a opção
escolhida pelo utilizador com a opção de 50 cêntimos

JEQ InseriuDinheiroPEPE ; Se a opção
escolhida pelo utilizador for a opção de 50 cêntimos, salta para a etiqueta
InseriuDinheiroPEPE

CMP R1, OpcaoSeguinteInserir1 ; Compara a opção
escolhida pelo utilizador com a opção de seguinte

JEQ InserirDinheiroPEPE2 ; Se a opção
escolhida pelo utilizador for a opção de seguinte, salta para a etiqueta InserirDinheiroPEPE2

CMP R1, OpcaoCancelarInserir1 ; Compara a opção
escolhida pelo utilizador com a opção de cancelar

JEQ VoltarConsultarCartao ; Se a opção
escolhida pelo utilizador for a opção de cancelar, salta para a etiqueta VoltarConsultarCartao

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
RotinaErro

JMP InserirDinheiroPEPE1 ; Salta para a
etiqueta InserirDinheiroPEPE1 novamente

InserirDinheiroPEPE2:

MOV R2, Display_InserirDinheiro2 ; R2 tem o endereço do
display de inserção de dinheiro

CALL MostraDisplay ;
Mostra o display de inserção de dinheiro

CALL LimpaPerifericos ; Limpa os
periféricos

LeOpcaoRecarregarPEPE2:

MOV R0, PER_EN ;
R0 tem o endereço do periférico de entrada
MOVB R1, [R0] ;
R1 tem o valor do periférico de entrada
CMP R1, 0 ;
Compara o valor do periférico de entrada com 0
JEQ LeOpcaoRecarregarPEPE2 ; Se o valor
do periférico de entrada for 0, repete o ciclo

MOV R5, Valor1Euro ;
R5 tem o valor de 1 euro
CMP R1, Opcao1EuroInserir2 ; Compara
a opção escolhida pelo utilizador com a opção de 1 euro
JEQ InserirDinheiroPEPE ; Se a opção
escolhida pelo utilizador for a opção de 1 euro, salta para a etiqueta InserirDinheiroPEPE

MOV R5, Valor2Euros ; R5 tem o
valor de 2 euros
CMP R1, Opcao2EuroInserir2 ; Compara
a opção escolhida pelo utilizador com a opção de 2 euros
JEQ InserirDinheiroPEPE ; Se a opção
escolhida pelo utilizador for a opção de 2 euros, salta para a etiqueta InserirDinheiroPEPE

MOV R5, Valor5Euros ; R5 tem o
valor de 5 euros
CMP R1, Opcao5EurosInserir2 ; Compara a opção
escolhida pelo utilizador com a opção de 5 euros
JEQ InserirDinheiroPEPE ; Se a opção
escolhida pelo utilizador for a opção de 5 euros, salta para a etiqueta InserirDinheiroPEPE

CMP R1, OpcaoSeguinteInserir2 ; Compara a opção
escolhida pelo utilizador com a opção de seguinte
JEQ InserirDinheiroPEPE3 ; Se a opção
escolhida pelo utilizador for a opção de seguinte, salta para a etiqueta InserirDinheiroPEPE3

CMP R1, OpcaoCancelarInserir2 ; Compara a opção
escolhida pelo utilizador com a opção de cancelar
JEQ VoltarConsultarCartao ; Se a opção
escolhida pelo utilizador for a opção de cancelar, salta para a etiqueta VoltarConsultarCartao

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
RotinaErro

JMP InserirDinheiroPEPE2 ; Salta para a
etiqueta InserirDinheiroPEPE2 novamente

InserirDinheiroPEPE3:

MOV R2, Display_InserirDinheiro3 ; R2 tem o endereço do
display de inserção de dinheiro

CALL MostraDisplay ;
Mostra o display de inserção de dinheiro

CALL LimpaPerifericos ; Limpa os
periféricos

LeOpcaoRecarregarPEPE3:

MOV R0, PER_EN ;
R0 tem o endereço do periférico de entrada

MOVB R1, [R0] ;
R1 tem o valor do periférico de entrada

CMP R1, 0 ;
Compara o valor do periférico de entrada com 0

JEQ LeOpcaoRecarregarPEPE3 ; Se o valor
do periférico de entrada for 0, repete o ciclo

MOV R5, Valor10Euros ; R5 tem o
valor de 10 euros

CMP R1, Opcao10EuroInserir3 ; Compara a opção
escolhida pelo utilizador com a opção de 10 euros

JEQ InseriuDinheiroPEPE ; Se a opção
escolhida pelo utilizador for a opção de 10 euros, salta para a etiqueta InseriuDinheiroPEPE

MOV R5, Valor20Euros ; R5 tem o
valor de 20 euros

CMP R1, Opcao20EuroInserir3 ; Compara a opção
escolhida pelo utilizador com a opção de 20 euros

JEQ InseriuDinheiroPEPE ; Se a opção
escolhida pelo utilizador for a opção de 20 euros, salta para a etiqueta InseriuDinheiroPEPE

CMP R1, OpcaoSeguinteInserir3 ; Compara a opção
escolhida pelo utilizador com a opção de seguinte

JEQ InserirDinheiroPEPE1 ; Se a opção
escolhida pelo utilizador for a opção de seguinte, salta para a etiqueta InserirDinheiroPEPE1

CMP R1, OpcaoCancelarInserir3 ; Compara a opção
escolhida pelo utilizador com a opção de cancelar

JEQ VoltarConsultarCartao ; Se a opção
escolhida pelo utilizador for a opção de cancelar, salta para a etiqueta VoltarConsultarCartao

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for nenhuma das opções anteriores, chama a rotina
RotinaErro

JMP InserirDinheiroPEPE3 ; Salta para a
etiqueta InserirDinheiroPEPE3 novamente

InseriuDinheiroPEPE:

MOV R2, Display_InseriuDinheiroPEPE ; R2 tem o endereço do
display de dinheiro inserido

CALL MostraDisplay ;
Mostra o display de dinheiro inserido

CALL LimpaPerifericos ; Limpa os
periféricos

CALL AdicionaStock ;
Chama a rotina AdicionaStock, que adiciona o dinheiro inserido ao stock da máquina

MOV R6, SaldoPEPEAtual ; R6 tem o
endereço do saldo do PEPE atual

MOV R7, [R6] ;
R7 tem o valor do saldo do PEPE atual

ADD R7, R5 ;
Adiciona o valor do dinheiro inserido ao saldo do PEPE atual

MOV [R6], R7 ;
Atualiza o saldo do PEPE atual

MOV R10, SaldoPEPEAtual ; R10 tem o
endereço do saldo do PEPE atual

MOV R6, NPEPEAtual ;
R6 tem o endereço do número do PEPE atual

MOV R11, [R6] ;
R11 tem o valor do número do PEPE atual

CALL AtualizarSaldoPEPE ; Chama a rotina
AtualizarSaldoPEPE, que atualiza o saldo do PEPE atual

CALL EscreverInseridoPEPE ; Chama a
rotina EscreverInseridoPEPE, que escreve o dinheiro inserido no display

LeOpcaoInseriuDinheiroPEPE:

MOV R0, PER_EN ;

R0 tem o endereço do periférico de entrada

MOVB R1, [R0] ;

R1 tem o valor do periférico de entrada

CMP R1, 0 ;

Compara o valor do periférico de entrada com 0

JEQ LeOpcaoInseriuDinheiroPEPE ; Se o valor do
periférico de entrada for 0, repete o ciclo

CMP R1, OpcaoContinuarInseriuDinheiroPEPE ; Compara a opção
escolhida pelo utilizador com a opção de continuar

JEQ InserirDinheiroPEPE1 ; Se a opção
escolhida pelo utilizador for a opção de continuar, salta para a etiqueta InserirDinheiroPEPE1

CALL RotinaErro ; Se a
opção escolhida pelo utilizador não for a opção de continuar, chama a rotina RotinaErro

JMP InseriuDinheiroPEPE ; Salta para a
etiqueta InseriuDinheiroPEPE novamente

----- Etiqueta

Stock

; Etiqueta que permite ao utilizador verificar o stock da máquina e autenticar-se para tal

;Verificação da Palavra

Passe-----

VerificacaoStock:

MOV R2, Display_VerificacaoStock ; R2 tem o endereço do
display para autenticação no stock

CALL MostraDisplay ; Mostra o
display de autenticação no stock
CALL LimpaPerifericos ; Limpa os
periféricos

LePalavraPasseStock:

MOV R0, PER_EN ; R0 tem o
endereço do botão opção
MOVB R1, [R0] ;
Leitura do periférico Opção
CMP R1, 0 ;
Compara o valor do botão opção com 0
JEQ LePalavraPasseStock ; Se o valor
do botão for 0, refaz a leitura

CMP R1, OpcaoConsultarVerificarStock ; Compara o valor do
botão opção com a opção de consultar stock
JEQ ConsultarVerificarStock ; Se o valor do
botão for igual à opção de consultar stock, salta para a etiqueta ConsultarVerificarStock

CMP R1, OpcaoCancelarVerificarStock ; Compara o valor
do botão opção com a opção de cancelar
JEQ VoltarInicioStock ; Se o valor
do botão for igual à opção de cancelar, salta para o menu principal

CALL RotinaErro ; Se o valor
do botão não for nenhuma das opções anteriores, chama a rotina RotinaErro
JMP VerificacaoStock ; Salta para
a etiqueta VerificacaoStock novamente

ConsultarVerificarStock:

MOV R6, 0 ;
R6 tem o valor 0
CALL VerificaPalavraPasseStock ; Chama a rotina
VerificaPalavraPasseStock, que verifica se a palavra-passe introduzida está correta
CMP R6, 1 ;
Compara o valor de R6 com 1, que quer dizer que a palavra-passe está correta
JEQ ConsultarStock ; Se o valor
de R6 for 1, salta para a etiqueta ConsultarStock

;Palavra-Passe

Incorreta-----

PalavraPasseIntroduzidaErrada:

MOV R2, Display_PalavraPasseErrada ; R2 tem o
endereço do display de palavra-passe errada
CALL MostraDisplay ; Mostra o
display de palavra-passe errada
CALL LimpaPerifericos ; Limpa os
periféricos

LeOpcaoPalavraPasseIntroduzidaErrada:

MOV R0, PER_EN ; R0 tem o
endereço do botão opção
MOVB R1, [R0] ;
Leitura do periférico Opção
CMP R1, 0 ;
Compara o valor do botão opção com 0
JEQ LeOpcaoPalavraPasseIntroduzidaErrada ; Se o valor do botão for
0, refaz a leitura

CMP R1, OpcaoVoltarErro ; Compara
o valor do botão opção com a opção de voltar
JEQ VerificacaoStock ; Se o valor do
botão for igual à opção de voltar, salta para a etiqueta VerificacaoStock

CALL RotinaErro ; Se o valor
do botão não for nenhuma das opções anteriores, chama a rotina RotinaErro
JMP PalavraPasseIntroduzidaErrada ; Salta para a etiqueta
PalavraPasseIntroduzidaErrada novamente

;Exposição do Stock da
máquina-----

ConsultarStock:

MOV R2, Display_ConsultarStock1 ; R2 tem o
endereço do display do menu de consultar stock
CALL MostraDisplay ; Mostra o
primeiro display de consultar stock
CALL LimpaPerifericos ; Limpa os
periféricos

MOV R6, 3 ;
R6 tem o valor 3 que é o número de valores a escrever no display
MOV R7, StockMoedas10 ; R7 tem o
endereço do primeiro valor a escrever no display
CALL EscreverStock ; Chama a
rotina EscreverStock, que escreve as quantidades dos valores monetários no stock da
maquina no display

LeOpcaoConsultarStock:

MOV R0, PER_EN ; R0 tem o
endereço do botão opção
MOVB R1, [R0] ;
Leitura do periférico Opção
CMP R1, 0 ;
Compara o valor do botão opção com 0
JEQ LeOpcaoConsultarStock ; Se o valor do
botão for 0, refaz a leitura

CMP R1, OpcaoSeguinteStock ; Compara
o valor do botão opção com a opção de seguinte
JEQ ConsultarStock2 ; Se o valor
do botão for igual à opção de seguinte, salta para a próxima página de stock

CMP R1, OpcaoSairStock ; Compara
o valor do botão opção com a opção de sair
JEQ VoltarInicioStock ; Se o valor
do botão for igual à opção de sair, salta para o menu principal

CALL RotinaErro ; Se o valor
do botão não for nenhuma das opções anteriores, chama a rotina RotinaErro
JMP ConsultarStock ; Salta para
a etiqueta ConsultarStock novamente

ConsultarStock2:

MOV R2, Display_ConsultarStock2 ; R2 tem o
endereço do display do menu de consultar stock
CALL MostraDisplay ; Mostra o
segundo display de consultar stock
CALL LimpaPerifericos ; Limpa os
periféricos
MOV R6, 3 ;
R6 tem o valor 3 que é o número de valores a escrever no display

MOV R7, StockMoedas1 ; R7 tem o endereço do primeiro valor a escrever no display

CALL EscreverStock ; Chama a rotina EscreverStock, que escreve as quantidades dos valores monetários no stock da maquina no display

LeOpcaoConsultarStock2:

MOV R0, PER_EN ; R0 tem o endereço do botão opção

MOVB R1, [R0] ; Leitura do periférico Opção

CMP R1, 0 ; Compara o valor do botão opção com 0

JEQ LeOpcaoConsultarStock2 ; Se o valor do botão for 0, refaz a leitura

CMP R1, OpcaoSeguinteStock ; Compara o valor do botão opção com a opção de seguinte

JEQ ConsultarStock3 ; Se o valor do botão for igual à opção de seguinte, salta para a próxima página de stock

CMP R1, OpcaoSairStock ; Compara o valor do botão opção com a opção de sair

JEQ VoltarInicioStock ; Se o valor do botão for igual à opção de sair, salta para o menu principal

CALL RotinaErro ; Se o valor do botão não for nenhuma das opções anteriores, chama a rotina RotinaErro

JMP ConsultarStock2 ; Salta para a etiqueta ConsultarStock2 novamente

ConsultarStock3:

MOV R2, Display_ConsultarStock3 ; R2 tem o endereço do display do menu de consultar stock

CALL MostraDisplay ; Mostra o terceiro display de consultar stock

CALL LimpaPerifericos ; Limpa os periféricos

MOV R6, 2 ; R6 tem o valor 2 que é o número de valores a escrever no display

MOV R7, StockNotas10 ; R7 tem o endereço do primeiro valor a escrever no display

CALL EscreverStock ; Chama a rotina EscreverStock, que escreve as quantidades dos valores monetários no stock da maquina no display

LeOpcaoConsultarStock3:

MOV R0, PER_EN ; R0 tem o endereço do botão opção

MOVB R1, [R0] ;
Leitura do periférico Opção

CMP R1, 0 ;
Compara o valor do botão opção com 0

JEQ LeOpcaoConsultarStock3 ; Se o valor do botão for 0, refaz a leitura

CMP R1, OpcaoSeguinteStock ; Compara o valor do botão opção com a opção de seguinte

JEQ ConsultarStock ; Se o valor do botão for igual à opção de anterior, salta para a página anterior de stock

CMP R1, OpcaoSairStock ; Compara o valor do botão opção com a opção de sair

JEQ VoltarInicioStock ; Se o valor do botão for igual à opção de sair, salta para o menu principal

CALL RotinaErro ; Se o valor do botão não for nenhuma das opções anteriores, chama a rotina RotinaErro

JMP ConsultarStock3 ; Salta para a etiqueta ConsultarStock3 novamente

VoltarInicioStock: ; Etiqueta que permite ao utilizador voltar ao menu principal usando um JMP para a etiqueta Ligado pois este tem alcance suficiente

JMP Ligado

Anexo D - Figuras dos Menus

MAQUINA VENDAS
METRO

1 - Compr ar
2 - Us ar Cart a o
3 - St o c k

Figura 1 - Menu Inicial

M E N U E S T A C A O
1 - E S T A C A O 1 : 1 . 5 0
2 - E S T A C A O 2 : 2 . 5 0
3 - E S T A C A O 3 : 4 . 0 0
4 - E S T A C A O 4 : 5 . 5 0
5 - E S T A C A O 5 : 7 . 0 0
6 - C a n c e l a r

Figura 2 - Menu de Comprar

- - - - -
E S C O L H E U B I L H E T E

D E S E J A E S C O L H E R
O U T R A E S T A C A O ?
1 - S i m 2 - N a o
- - - - -

Figura 3 - Menu Confirmação Escolha


```

- - - - -
      P R E C O   T O T A L
- - - - -
              1 . 5 0
1   -   C o n t i n u a r
2   -   C a n c e l a r
- - - - -

```

Figura 4 - Menu Mostra Preço Total

```

      I N S E R C A O
      D E   D I N H E I R O
1   -   1 0   C e n t
2   -   2 0   C e n t
3   -   5 0   C e n t
5   -   S e g u i n t e
7   -   C a n c e l a r

```

Figura 5 - Menus Inserir Dinheiro

```

- - - - -
      I N S E R I U :      1 . 0 0
      F A L T A :         0 . 5 0

1   -   C o n t i n u a r
- - - - -

```

Figura 6 - Menu Inseriu Dinheiro

```

      P E P E   G E R A D O
              0 0 1
T o t a l :           1 . 5 0
I n s e r i d o :     2 . 0 0
T r o c o :           0 . 5 0

1   -   C o n t i n u a r

```

Figura 7 - Menu Talão

P A R A B E N S
 V I A G E M G R A T U I T A !

 V A L O R D A C O M P R A
 C A R R E G A D O
 N O C A R T A O
 1 - C o n t i n u a r

Figura 8 - Menu Primeira Compra

 R E T I R E O S E U
 T R O C O !
 1 0 C e n t . . . 0
 2 0 C e n t . . . 0
 5 0 C e n t . . . 1
 5 - S e g u i n t e
 7 - S a i r

Figura 9 - Menus Devolver Troco

 O B R I G A D O
 P E L A S U A
 C O M P R A !

 7 - C o n t i n u a r

Figura 10 - Menu Agradecimento

 I N T R O D U Z A N .
 P E P E

 X X

 1 - C o n t i n u a r
 5 - C a n c e l a r

Figura 11 - Menu Usar Cartão

```

- - - - - E R R O - - - - -

      P E P E
    i n e x i s t e n t e

      7 - V o l t a r
- - - - - E R R O - - - - -

```

Figura 12 - Menu PEPE Não Encontrado

```

      P E P E 1

S a l d o :    0 1 . 5 0

1 - C o m p r a r
2 - R e c a r r e g a r
7 - V o l t a r

```

Figura 13 - Menu Consultar Cartão

```

- - - - - E R R O - - - - -

      S A L D O P E P E
    I N S U F I C I E N T E

      7 - V o l t a r
- - - - - E R R O - - - - -

```

Figura 14 - Menu Saldo Insuficiente

```

      S T O C K

P A L A V R A - P A S S E :
      * * * * *

1 - C o n s u l t a r
7 - C a n c e l a r

```

Figura 15 - Menu Stock

```

- - - - E R R O - - - -
P A L A V R A - P A S S E
      I N S E R I D A
      I N C O R R E T A

7   -   V o l t a r
- - - - E R R O - - - -

```

Figura 16 - Menu Palavra-Passe Incorreta

```

      C O N S U L T A
      S T O C K
1 0   C e n t . . .   1 0
2 0   C e n t . . .   1 0
5 0   C e n t . . .    9
5   -   S e g u i n t e
7   -   S a i r

```

Figura 17 - Menus Consultar Stock