

Arquitetura de Computadores

2020/2021



2ª Projeto

Balança com calculadora de Macros

Docentes:

- João Dionísio Simões Barros
- Nuno Fábio Gomes Camacho Ferreira
- Pedro Miguel Pestana Camacho
- Sofia Isabel Silva

Membros do grupo:

- Hugo Rocha nº 2046019;
- Sérgio Oliveira nº 2049719;

ÍNDICE

Introdução.....	3
Objetivo.....	3
Desenvolvimento.....	3
Endereçamento.....	3
Periféricos.....	3
Rotinas.....	3
Conclusão.....	4
Anexo A (fluxogramas).....	4
Anexo B (código em linguagem assembly).....	4

Introdução

Este trabalho prático foi realizado no âmbito curricular de Arquitetura de Computadores, na qual foi-nos proposto realizar uma Balança com Calculadora de macros, em linguagem Assembly e com as respetivas instruções do processador PEPE. Para realizar o trabalho necessitámos de recorrer ao software disponibilizado pelos docentes desta UC.

De início, foi necessário perceber o funcionamento da arquitetura do PEPE, uma vez que este processador é diferente do PEPE-8 do projeto anterior. De modo a facilitar a interpretação das instruções e conceitos do programa foram elaborados fluxogramas que ajudam na compreensão do mesmo.

Objetivo

Este trabalho prático tem como propósito criar um programa que simule uma balança que calcule os macronutrientes e calorias de um dado alimento, bem como a habilidade de poder estabelecer metas diárias e verificar se elas foram ou não ultrapassadas.

A balança tem uma precisão de 1g, e um limite de 2500g. Caso o peso ultrapasse os 2500g leva será efetuado o seu reset, voltando a 0g. Irá permitir selecionar o alimento em questão, uma vez que os macronutrientes variam de alimento para alimento. Com os macronutrientes calculados estes têm de sair adicionados ao registo diário, que mostra o total de calorias e total de cada um dos macronutrientes.

A balança contém também uma funcionalidade que permite limpar o total diário dos macronutrientes já inseridos.

Utilização

A balança suporta escolhas erradas por parte do utilizador, mostrando um ecrã de erro e continuando sem problemas a execução do programa.

No menu balança, é permitido mudar o alimento. Basta clicar em CHANGE, e logo de seguida escolher o número do alimento e clicar em OK. Também é possível ir verificando os vários alimentos, clicando repetidamente no botão CHANGE, para ir mostrando os vários alimentos da tabela.

Para sair do menu do total diário, basta clicar no OK.

Para mudar a meta diária de um dado macronutriente, basta no respetivo menu inserir o peso máximo que pretende escolher, selecionar o macronutriente e clicar em OK.

Se a meta diária de um determinado macronutriente for excedida, foi implementado um algoritmo onde o seu valor é mostrado a 0.

Desenvolvimento

Foi desenvolvida uma rotina que permite arredondar as calorias, caso o valor do peso não seja múltiplo de 100. Foi também desenvolvida uma rotina que permite converter um valor de hexadecimal para ASCII, de forma a mostrar o valor corretamente ao utilizador, e outra rotina que calcula as calorias de um certo número de macronutrientes.

Foram desenvolvidas duas rotinas, talvez as mais interessantes do programa, que permitem dar overwrite a uma linha específica do display. Desta maneira, após ser inserido o menu do peso por exemplo, é colocado o valor peso numa linha escolhida pelo programador.

Endereçamento

Tivemos o cuidado de verificar se as tabelas de alimentos e os menus não ficavam no mesmo sítio da memória que as instruções. Os valores da meta dos vários macros, e do total, são também guardados em memória, tendo também este cuidado de não haver sobreposição.

Foi escolhido um endereço para o display e outros periféricos relativamente perto do 0, para ficar logo no início da memória, para permitir um mais rápido debug.

Os endereços dos vários botões são os seguintes:

B_ON_OFF	EQU 0020H
SEL_NR_MENU	EQU 0022H
B_OK	EQU 0024H
B_CHANGE	EQU 0026H
PESO	EQU 0028H

Conclusão

Este trabalho permitiu-nos criar uma balança que calcula os macronutrientes, no software disponibilizado pelos docentes desta unidade curricular, numa linguagem de baixo nível, que é o assembly. Este projeto demonstrou ser muito trabalhoso, pois não só foi necessário algum tempo de estudo, como também algum tempo a compreender o simulador, e alguns detalhes relativos a algumas das instruções.

O debug do programa foi relativamente difícil, ainda mais devido a um bug do simulador que foi encontrado (onde a linha 0192H era pura e simplesmente ignorada). A habilidade de poder efetuar breakpoints nas várias rotinas ajudou imensamente neste processo.

Contudo, os requisitos foram todos eles cumpridos, e de forma robusta, com exceção da deteção do overflow dos macronutrientes.

Para concluir, o programa encontra-se apresentado de uma forma clara e sem erros. A realização deste trabalho prático, permitiu-nos consolidar os conhecimentos adquiridos tanto na parte prática, como na parte teórica da unidade curricular, e ampliar o nosso conhecimento na programação em linguagem assembly, e nas instruções do PEPE em particular.

É vivamente recomendado o uso do VsCode com a extensão Amiga-Assembly para uma fácil visualização do código. Todo o código foi também formatado usando esta extensão.

Anexo A (fluxogramas)

Anexo B (código em linguagem assembly)

; Projecto 2 Arquitetura de Computadores - Balança com calculadora de Macros

; Note - It's not peripherals, it's peripherals.

; Note - Instrução 0192H não executa, bug do simulador.

; Endereçamento do Programa

; Pula para o bloco de instruções -> 0 ou 0x0000

; Periféricos -> 32 - 255 ou 0x0020 - 0x00FF

; Bloco de instruções -> 256 - 1023 ou 0x0100 - 0x03FF

; Menus -> 2048 - 4095 ou 0x0800 - 0x0FFF

; Tabela de alimentos -> 4096 - 8191 ou 0x1000 - 0x1FFF

;

; Periféricos de Input

; É tomada a escolha que os periféricos são todos de palavra, por razões de "Future Proofing"

B_ON_OFF EQU 0020H ; Endereço botão on/off

SEL_NR_MENU EQU 0022H ; Endereço botão
para selecionar a opção nos menus

B_OK EQU 0024H ; Endereço botão validar
escolha

B_CHANGE EQU 0026H ; Endereço botão
change

PESO EQU 0028H ; Endereço do periférico
peso

;

; Display

; O display tem dimensões 7x16 (7 linhas de 16 bytes)

; 7x16 = 112

; 112 = 0x70

DisplayBeginning EQU 0080H ; Endereço onde
começa o display

DisplayEnd EQU 00EFH ; Endereço onde acaba o
display. $0x0080 + 0x006F = 0x00EF$

; Number

DisplayNumber0 EQU 0040H

DisplayNumber1 EQU 0041H

DisplayNumber2 EQU 0042H

DisplayNumber3 EQU 0043H

;

; Calorias

; Total

CALORIAS EQU 0030H

PROTEINA EQU 0032H

HIDRATOS EQU 0034H

GORDURA EQU 0036H

; Meta

META_CALORIAS EQU 0038H

META_PROTEINA EQU 003AH

META_HIDRATOS EQU 003CH

META_GORDURA EQU 003EH

; Diferença Meta-Calorias

DIFERENCA_CALORIAS EQU 0044H

DIFERENCA_PROTEINA EQU 0046H

DIFERENCA_HIDRATOS EQU 0048H

DIFERENCA_GORDURA EQU 004AH

;

; Stack Pointer

StackPointer EQU 1FFEh ; Endereço do Stack
Pointer

;

; Tabela de Alimentos

PLACE 1000H

TableAveia	:
STRING	"00- AVEIA "
STRING	"011 056 007 -"

TablePaoForma	:
STRING	"01- PAO DE FORMA"
STRING	"009 042 003 -"

TableBatata	:
STRING	"02- BATATA "
STRING	"003 019 000 -"

TableArroz	:
STRING	"03- ARROZ "
STRING	"007 025 000 -"

TableFeijao	:
STRING	"04- FEIJAO "

STRING "010 013 000 -"

TableLegumes :

STRING "05- LEGUMES "

STRING "003 007 000 -"

TableTomate :

STRING "06- TOMATE "

STRING "001 003 000 -"

TableBanana :

STRING "07- BANANA "

STRING "001 023 000 -"

TableLaranja :

STRING "08- LARANJA "

STRING "001 012 000 -"

TableMaca :

STRING "09- MACA "

STRING "001 014 000 -"

TableKiwi :

STRING "10- KIWI "

STRING "001 015 000 -"

TableBolachaChoc :

STRING "11- BOLACHA CHOC"

STRING "009 059 022 -"

TablePizza :

STRING	"12- PIZZA "
STRING	"013 025 009 -"
TableAmendoas	:
STRING	"13- AMENDOAS "
STRING	"025 006 055 -"
TableLinhacas	:
STRING	"14- LINHACAS "
STRING	"018 034 036 -"
TableAzeite	:
STRING	"15- AZEITE "
STRING	"000 000 100 -"
TableLMagro	:
STRING	"16- LEITE MAGRO "
STRING	"003 004 000 -"
TableWhey	:
STRING	"17- WHEY "
STRING	"080 008 004 -"
TableSalmao	:
STRING	"18- SALMAO "
STRING	"021 000 015 -"
TablePescada	:
STRING	"19- PESCADA "
STRING	"020 000 001 -"

TableAtum :
STRING "20- ATUM "
STRING "025 000 002 -"

TablePorco :
STRING "21- PORCO "
STRING "022 000 015 -"

TableFrango :
STRING "22- FRANGO "
STRING "025 000 004 -"

TablePeru :
STRING "23- PERU "
STRING "028 000 001 -"

TableOvo :
STRING "24- OVO "
STRING "007 000 005 -"

TableQueijo :
STRING "25- QUEIJO "
STRING "028 000 013 -"

;

; Menus

PLACE 2000H

```

GUIMenuMain          :
STRING                " MENU INICIAL "
STRING                "
STRING                "1- BALANCA  "
STRING                "2- TOTAL DIARIO "
STRING                "3- META DIARIA "
STRING                "4- RESET  "
STRING                "

```

```

GUIMenuScale         :
STRING                " PESO  "
STRING                "
STRING                "
STRING                " ALIMENTO "
STRING                "
STRING                "
STRING                "

```

```

GUIMenuChoiceError   :
STRING                " ERRO  "
STRING                "
STRING                " OPCA0  "
STRING                " INVALIDA "
STRING                "
STRING                " PRIMA OK PARA "
STRING                " CONTINUAR "

```

```

GUIMenuDailyTotal    :
STRING                "TOTAL DIA(P,H,G)"
STRING                "
STRING                "

```

STRING " "

STRING " "

STRING " CALORIAS "

STRING " "

GUIMenuDailyGoal :

STRING " META DIA "

STRING " "

STRING "1- ALTERAR "

STRING "2- VISUALIZAR "

STRING " "

STRING " "

STRING " "

GUIMenuDailyGoalChange :

STRING "META DIA-ALTERAR"

STRING " "

STRING "1- PROTEINA "

STRING "2- HIDRATOS "

STRING "3- GORDURA "

STRING "4- CALORIAS "

STRING " "

GUIMenuDailyGoalSee :

STRING " META DIA - VER "

STRING " P, H, G, C "

STRING " "

STRING " "

STRING " "

STRING " "

STRING " "

```

GUIMenuReset          :
STRING                " MENU RESET "
STRING                "
STRING                "1- RESET  "
STRING                "2- MAIN MENU "
STRING                "
STRING                "
STRING                "

```

;

; Instruções

PLACE 0000H

```

JMP                  Startup

```

PLACE 0100H

MemoryVariablesResetCall:

```

MOV                  R0, 2
MOV                  R1, 0038H

MOV                  [R1], R0
ADD                  R1, R0

MOV                  [R1], R0
ADD                  R1, R0

```

MOV [R1], R0

ADD R1, R0

RET

DisplayResetCall: ; (Display Beginning, DisplayEnd + 1)

CaraterVazio EQU 2020H ; Carater vazio usado
para limpar o ecrã

MOV R3, CaraterVazio ; Guardar em R3 o carater
vazio

MOV [R0], R3 ; Guardar no display o carater
vazio

;ADD R2, 2 ; Pula para a próxima palavra
do menu

ADD R0, 2 ; Pula para a próxima palavra do
display

; Se ainda não chegou ao fim do display, começar a call de novo

CMP R0, R1

JNE DisplayResetCall

; Se chegou ao fim do display, retornar

RET

PeripheralsResetCall:

; Mover os endereços dos vários periféricos para os registos

MOV R0, B_ON_OFF ; Guardar Endereço botão
on/off

MOV R1, SEL_NR_MENU ; Guardar Endereço
botão para seleccionar a opção nos menus

```
MOV          R2, B_OK          ; Guardar Endereço botão  
validar escolha  
  
MOV          R3, B_CHANGE      ; Guardar Endereço  
botão change  
  
MOV          R4, PESO          ; Guardar Endereço do  
periférico peso  
  
MOV          R5, 0             ; Guardar valor a usar nos  
vários resets dos periféricos
```

; Mover 0 para todos os periféricos, e 0 ASCII para o peso, fazendo o seu reset

```
MOVB         [R0], R5  
MOVB         [R1], R5  
MOVB         [R2], R5  
MOVB         [R3], R5  
MOV          [R4], R5
```

; Retornar da sub-rotina

```
RET
```

PrepareDisplayCall: ; (No Input), (DisplayBeginning, (DisplayEnd + 1)

; Uses R0 - R1

```
MOV          R0, DisplayBeginning ; Guardar em R0 o  
endereço do início do display
```

```
MOV          R1, DisplayEnd       ; Guardar em R1 o  
endereço do final do display
```

```
ADD          R1, 1                ; Adicionar 1 ao display end,  
quando o iterador do display chegar a este valor a rotina vai ser indicada a parar
```

```
RET
```

OverwriteDisplayCall: ; ([DisplayBeginning], LineToBeOverwritten, [ContentToOverwrite],
BytesAlreadyOverwritten)

MOV R4, 16 ; Número de bytes que uma linha do display tem

MUL R1, R4 ; Transforma a LineToBeOverwritten no padding que será necessário dar ao [DisplayBeginning]

ADD R0, R1 ; Muda o [DisplayBeginning] para o início da linha do display onde se quer dar overwrite

OverwriteDisplayPrepared :

CMP R3, R4 ; Verifica se já foram dados overwrite a 16 bytes, ou seja, à linha inteira

JNE BytesNotAllOverwritten

; Se já foi dado overwrite a todos os bytes da linha

RET

; Se ainda não foi dado overwrite a todos os bytes da linha

BytesNotAllOverwritten :

MOV R5, [R2] ; Mover valor ContentToOverwrite para R5

MOV [R0], R5 ; Mover ContentToOverwrite que se quer dar overwrite para o display

ADD R0, 2

ADD R2, 2

ADD R3, 2

JMP OverwriteDisplayPrepared

ConvertMemoryToASCII: ; ([NumberToConvert])

; Usado para converter os endereços em memória (PROTEINA,HIDRATOS,GORDURA) para ASCII

MOV R0, [R0] ; Mover valor NumberToConvert para R0

ConvertMemoryToASCIIFirstNumber :

MOV	R1, R0	; Cria cópia do valor
NumberToConvert em R1		
MOV	R2, 1000	; Move 1000 para o R2
DIV	R1, R2	; Guardar resultado da divisão
inteira em R1		
MOV	R2, 0030H	; Move 48 para o R2
ADD	R1, R2	; Converter cópia do resultado
para ASCII		
MOV	R2, DisplayNumber0	; Move endereço do
DisplayNumber0 para o R2		
MOVB	[R2], R1	; Mover número ASCII para o
DisplayNumber0		
MOV	R2, 1000	; Move 1000 para o R2
MOD	R0, R2	; Guardar resto da divisão em
R0		

ConvertMemoryToASCIISecondNumber :

MOV	R1, R0	; Cria cópia do valor
NumberToConvert em R1		
MOV	R2, 100	; Move 1000 para o R2
DIV	R1, R2	; Guardar resultado da divisão
inteira em R1		
MOV	R2, 0030H	; Move 48 para o R2
ADD	R1, R2	; Converter cópia do resultado
para ASCII		
MOV	R2, DisplayNumber1	; Move endereço do
DisplayNumber0 para o R2		
MOVB	[R2], R1	; Mover número ASCII para o
DisplayNumber0		
MOV	R2, 100	; Move 1000 para o R2
MOD	R0, R2	; Guardar resto da divisão em
R0		

ConvertMemoryToASCIIThirdNumber :

MOV	R1, R0	; Cria cópia do valor
NumberToConvert em R1		
MOV	R2, 10	; Move 1000 para o R2
DIV	R1, R2	; Guardar resultado da divisão
inteira em R1		
MOV	R2, 0030H	; Move 48 para o R2
ADD	R1, R2	; Converter cópia do resultado
para ASCII		
MOV	R2, DisplayNumber2	; Move endereço do
DisplayNumber0 para o R2		
MOVB	[R2], R1	; Mover número ASCII para o
DisplayNumber0		
MOV	R2, 10	; Move 1000 para o R2
MOD	R0, R2	; Guardar resto da divisão em
R0		

ConvertMemoryToASCIIFourthNumber :

MOV	R1, R0	; Cria cópia do valor
NumberToConvert em R1		
MOV	R2, 1	; Move 1000 para o R2
DIV	R1, R2	; Guardar resultado da divisão
inteira em R1		
MOV	R2, 0030H	; Move 48 para o R2
ADD	R1, R2	; Converter cópia do resultado
para ASCII		
MOV	R2, DisplayNumber3	; Move endereço do
DisplayNumber0 para o R2		
MOVB	[R2], R1	; Mover número ASCII para o
DisplayNumber0		

RET

OverwriteDisplayFourBytesCall: ; ([DisplayBeginning], LineToBeOverwritten,
[ContentToOverwrite], BytesAlreadyOverwritten)

MOV R4, 16 ; Número de bytes que uma linha do display tem

MUL R1, R4 ; Transforma a LineToBeOverwritten no padding que será necessário dar ao [DisplayBeginning]

ADD R0, R1 ; Muda o [DisplayBeginning] para o início da linha do display onde se quer dar overwrite

OverwriteDisplayFourBytesPrepared :

MOV R4, 4

CMP R3, R4 ; Verifica se já foram dados overwrite a 4 bytes

JNE OverwriteDisplayFourBytesBytesNotAllOverwritten

; Se já foi dado overwrite a todos os bytes da linha

RET

; Se ainda não foi dado overwrite a todos os bytes da linha

OverwriteDisplayFourBytesBytesNotAllOverwritten :

MOV R5, [R2] ; Mover valor ContentToOverwrite para R5

MOV [R0], R5 ; Mover ContentToOverwrite que se quer dar overwrite para o display

ADD R0, 2

ADD R2, 2

ADD R3, 2

JMP OverwriteDisplayFourBytesPrepared

CalculateCalories:

MOV R0, 0 ; R0 guarda a soma das calorias dos macronutrientes

MOV R1, PROTEINA

MOV R1, [R1] ; Move valor proteina para R1

MOV	R2, 4	
MUL	R1, R2	; Calcula calorias da proteina
ADD	R0, R1	; Adiciona ao total de calorias
MOV	R1, HIDRATOS	
MOV R1	R1, [R1]	; Move valor de hidratos para R1
MOV	R2, 4	
MUL	R1, R2	; Calcula calorias dos hidratos
ADD	R0, R1	; Adiciona ao total de calorias
MOV	R1, GORDURA	
MOV R1	R1, [R1]	; Move valor de gordura para R1
MOV	R2, 9	
MUL	R1, R2	; Calcula calorias da gordura
ADD	R0, R1	; Adiciona ao total de calorias
MOV	R1, CALORIAS	
MOV memória	[R1], R0	; Move total de calorias para a memória
RET		

RoundMacros: ; (QuantityIn100) (TotalAmount)

; Uses R0 - R2

; Formula = TotalAmount * QuantityIn100

; Example: 123 grams * 11 proteina = 1353

MUL R0, R1 ; Guardar resultado de
TotalAmount * QuantityIn100 para R0

MOV R1, R0 ; Copiar resultado de
TotalAmount * QuantityIn100 para R1

MOV R2, 100

MOD R1, R2 ; 1253 % 100 = 53

; Verifica se o módulo é maior que 50

MOV R2, 50

CMP R1, R2

JGE RoundUp

; Se o módulo é menor que 50

MOV R2, 100

DIV R0, R2

RET

; Se o módulo é maior ou igual a 50

RoundUp :

MOV R2, 100

DIV R0, R2

MOV R2, 1

ADD R0, 1

RET

Startup:

;MOV SP, StackPointer ; Guardar o endereço do Stack Pointer no
registo SP

CALL PrepareDisplayCall ; Preparar o display para
ser limpo

CALL DisplayResetCall ; Chamar a rotina que
limpa o display

```
CALL                PeripheralsResetCall                ; Chamar a rotina que
limpa os periféricos

;CALL              CaloryResetCall                ; Chama a rotina que dá reset às calorias
gravadas em memória

; B_ON_OFF

CALL                CheckTurnOnCall                ; Verifica continuamente
se o botão de ligar foi pressionado

; Menu Input

CALL                MenuMainCall                ; Executa a call MainMenu
após o botão de ligar ser pressionado

JMP                Startup
```

CheckTurnOnCall:

```
MOV                R0, B_ON_OFF                ; Guardar o endereço de
B_ON_OFF em R0

MOV                R1, [R0]                ; Escrever o valor de
B_ON_OFF em R1

CMP                R1, 1                ; Comparar se o B_ON_OFF está
igual a 1 (ligado)

JNE                CheckTurnOnCall                ; Se B_ON_OFF estiver
desligado, volta a comparar até passar a ligado

RET
```

DisplayMenuCall: ; (Display Beginning, DisplayEnd + 1, [MenuToDisplay])

; Uses R0 - R3

```
MOV                R3, [R2]                ; Guardar em R3 uma palavra
do menu a imprimir no display

MOV                [R0], R3                ; Guardar no display o valor
escrito em R3

ADD                R2, 2                ; Pula para a próxima palavra do
menu
```

ADD R0, 2 ; Pula para a próxima palavra do display

; Se ainda não chegou ao fim do display, começar a call de novo

CMP R0, R1

JNE DisplayMenuCall

; Se chegou ao fim do display, retornar

RET

MenuChangeFoodCall: ; ((), (), (), (), (), (), (), TableNumber)

; R4 Guarda o número de caracteres da tabela de cada alimento, para ser usado como padding

; R5 Guarda cópia do TableNumber, a original tem que se manter para poder ser incrementada quando é clicado no B_CHANGE

; R6 não é alterado, mas MenuScaleCall chama esta rotina, e R6 é um dos seus parâmetros

; R7 - AlimentoAtual é alterado quando é selecionado outro

; TODO: After cycling trough all foods, reset back to the first

; Display

CALL PrepareDisplayCall ; Preparar ecrã para mostrar o menu balança

MOV R4, 32 ; Guarda em R4 o número de caracteres da tabela de cada alimento, para ser usado como padding

MOV R5, R8 ; Guarda cópia do TableNumber em R5, a original tem que se manter para poder ser incrementada quando é clicado no B_CHANGE

MUL R5, R4 ; Constrói o padding que será usado para aceder ao endereço correto (Cópia TableNumber * Nº Carateres Alimento)

MOV R2, TableAveia ; Guarda em R2 o endereço da primeira tabela

ADD R2, R5 ; Adiciona o padding ao endereço da primeira tabela, obtendo a tabela que se quer aceder

CALL DisplayMenuCall ; Mostrar Menu balanço no Display

MenuChangeFoodDisplayReady :

; Verifica se B_CHANGE for pressionado

MOV R0, B_CHANGE

MOV R1, [R0] ; Move o valor do B_CHANGE para R1

CMP R1, 1

JNE MenuChangeFoodChangeNotPressed

; Se o botão B_CHANGE for pressionado

MOV R0, B_CHANGE

MOV R1, 0

MOV [R0], R1 ; Reset do valor do B_CHANGE

ADD R8, 1 ; Passa como parâmetro o nº da tabela do alimento seguinte

JMP MenuChangeFoodCall

; Se o botão B_CHANGE não for pressionado

MenuChangeFoodChangeNotPressed :

MOV R2, SEL_NR_MENU

MOV R0, [R2] ; Move valor SEL_NR_MENU para R0

MOV R2, B_OK

MOV R1, [R2] ; Move valor B_OK para R1

```
CMP                                R1, 1  
  
JNE                                MenuChangeFoodDisplayReady      ; Se o botão  
change não foi clicado, e não foi selecionado nenhum alimento,  
  
;                                não há necessidade de voltar a dar refresh  
no ecrã, sendo só necessário verificar os inputs.
```

```
; Se o botão B_OK foi pressionado
```

```
ChoiceAveia                        :  
  
CMP                                R0, 0  
  
JNE                                ChoicePaoForma  
  
MOV                                R7, 0  
  
  
; Reset  
  
MOV                                R0, 0  
  
MOV                                R1, SEL_NR_MENU  
  
MOV                                [R1], R0                        ; Reset do SEL_NR_MENU  
  
MOV                                R1, B_OK  
  
MOV                                [R1], R0                        ; Reset do B_OK  
  
RET
```

```
ChoicePaoForma                    :  
  
CMP                                R0, 1  
  
JNE                                ChoiceBatata  
  
MOV                                R7, 1  
  
  
; Reset  
  
MOV                                R0, 0  
  
MOV                                R1, SEL_NR_MENU  
  
MOV                                [R1], R0                        ; Reset do SEL_NR_MENU  
  
MOV                                R1, B_OK
```

```
MOV          [R1], R0          ; Reset do B_OK
RET
```

```
ChoiceBatata      :
CMP            R0, 2
JNE            ChoiceArroz
MOV            R7, 2

; Reset
MOV            R0, 0
MOV            R1, SEL_NR_MENU
MOV            [R1], R0        ; Reset do SEL_NR_MENU
MOV            R1, B_OK
MOV            [R1], R0        ; Reset do B_OK
RET
```

```
ChoiceArroz       :
CMP            R0, 3
JNE            ChoiceFeijao
MOV            R7, 3

; Reset
```

```
MOV            R0, 0
MOV            R1, SEL_NR_MENU
MOV            [R1], R0                ; Reset do SEL_NR_MENU
MOV            R1, B_OK
MOV            [R1], R0                ; Reset do B_OK
RET
```

```
ChoiceFeijao      :
CMP            R0, 4
JNE            ChoiceLegumes
MOV            R7, 4

; Reset
MOV            R0, 0
MOV            R1, SEL_NR_MENU
MOV            [R1], R0                ; Reset do SEL_NR_MENU
MOV            R1, B_OK
MOV            [R1], R0                ; Reset do B_OK
RET
```

```
ChoiceLegumes     :
CMP            R0, 5
```

```
JNE          ChoiceTomate
MOV          R7, 5

; Reset
MOV          R0, 0
MOV          R1, SEL_NR_MENU
MOV          [R1], R0          ; Reset do SEL_NR_MENU
MOV          R1, B_OK
MOV          [R1], R0          ; Reset do B_OK
RET
```

```
ChoiceTomate      :
MOV          R1, 6
CMP          R0, R1
JNE          ChoiceBanana
MOV          R7, R1

; Reset
MOV          R0, 0
MOV          R1, SEL_NR_MENU
MOV          [R1], R0          ; Reset do SEL_NR_MENU
MOV          R1, B_OK
MOV          [R1], R0          ; Reset do B_OK
RET
```

```
ChoiceBanana          :  
MOV                    R1, 7  
CMP                    R0, R1  
JNE                    ChoiceLaranja  
MOV                    R7, R1  
  
; Reset  
MOV                    R0, 0  
MOV                    R1, SEL_NR_MENU  
MOV                    [R1], R0          ; Reset do SEL_NR_MENU  
MOV                    R1, B_OK  
MOV                    [R1], R0          ; Reset do B_OK  
RET
```

```
ChoiceLaranja         :  
MOV                    R1, 8  
CMP                    R0, R1  
JNE                    ChoiceMaca  
MOV                    R7, R1
```

```
; Reset
```

```
MOV            R0, 0
MOV            R1, SEL_NR_MENU
MOV            [R1], R0                ; Reset do SEL_NR_MENU
MOV            R1, B_OK
MOV            [R1], R0                ; Reset do B_OK
RET
```

```
ChoiceMaca      :
MOV            R1, 9
CMP            R0, R1
JNE            ChoiceKiwi
MOV            R7, R1
```

; Reset

```
MOV            R0, 0
MOV            R1, SEL_NR_MENU
MOV            [R1], R0                ; Reset do SEL_NR_MENU
MOV            R1, B_OK
MOV            [R1], R0                ; Reset do B_OK
RET
```

```
ChoiceKiwi      :
MOV            R1, 10
CMP            R0, R1
JNE            ChoiceBolachaChoc
```

```
MOV                R7, R1

; Reset
MOV                R0, 0
MOV                R1, SEL_NR_MENU
MOV                [R1], R0                ; Reset do SEL_NR_MENU
MOV                R1, B_OK
MOV                [R1], R0                ; Reset do B_OK
RET
```

```
ChoiceBolachaChoc      :
MOV                R1, 11
CMP                R0, R1
JNE                ChoicePizza
MOV                R7, R1

; Reset
MOV                R0, 0
MOV                R1, SEL_NR_MENU
MOV                [R1], R0                ; Reset do SEL_NR_MENU
MOV                R1, B_OK
MOV                [R1], R0                ; Reset do B_OK
RET
```



```
ChoicePizza          :
MOV                   R1, 12
CMP                   R0, R1
JNE                   ChoiceAmendoas
MOV                   R7, R1

; Reset
MOV                   R0, 0
MOV                   R1, SEL_NR_MENU
MOV                   [R1], R0           ; Reset do SEL_NR_MENU
MOV                   R1, B_OK
MOV                   [R1], R0           ; Reset do B_OK
RET
```

```
ChoiceAmendoas       :
MOV                   R1, 13
CMP                   R0, R1
JNE                   ChoiceLinhacas
MOV                   R7, R1

; Reset
MOV                   R0, 0
MOV                   R1, SEL_NR_MENU
MOV                   [R1], R0           ; Reset do SEL_NR_MENU
MOV                   R1, B_OK
MOV                   [R1], R0           ; Reset do B_OK
```

RET

ChoiceLinhacas :

MOV R1, 14

CMP R0, R1

JNE ChoiceAzeite

MOV R7, R1

; Reset

MOV R0, 0

MOV R1, SEL_NR_MENU

MOV [R1], R0 ; Reset do SEL_NR_MENU

MOV R1, B_OK

MOV [R1], R0 ; Reset do B_OK

RET

ChoiceAzeite :

MOV R1, 15

CMP R0, R1

JNE ChoiceLMagro

MOV R7, R1

; Reset

MOV R0, 0

MOV R1, SEL_NR_MENU

MOV [R1], R0 ; Reset do SEL_NR_MENU

```
MOV          R1, B_OK
MOV          [R1], R0          ; Reset do B_OK
RET
```

```
ChoiceLMagro      :
MOV          R1, 16
CMP         R0, R1
JNE         ChoiceWhey
MOV         R7, R1
```

; Reset

```
MOV          R0, 0
MOV          R1, SEL_NR_MENU
MOV          [R1], R0          ; Reset do SEL_NR_MENU
MOV          R1, B_OK
MOV          [R1], R0          ; Reset do B_OK
RET
```

```
ChoiceWhey      :
MOV          R1, 17
CMP         R0, R1
JNE         ChoiceSalmao
MOV         R7, R1
```

; Reset

```
MOV          R0, 0
MOV          R1, SEL_NR_MENU
MOV          [R1], R0          ; Reset do SEL_NR_MENU
```

```
MOV          R1, B_OK
MOV          [R1], R0          ; Reset do B_OK
RET
```

```
ChoiceSalmao      :
MOV          R1, 18
CMP          R0, R1
JNE          ChoicePescada
MOV          R7, R1

; Reset
MOV          R0, 0
MOV          R1, SEL_NR_MENU
MOV          [R1], R0          ; Reset do SEL_NR_MENU
MOV          R1, B_OK
MOV          [R1], R0          ; Reset do B_OK
RET
```

```
ChoicePescada     :
MOV          R1, 19
CMP          R0, R1
JNE          ChoiceAtum
MOV          R7, R1
```

; Reset

```
MOV          R0, 0
MOV          R1, SEL_NR_MENU
MOV          [R1], R0          ; Reset do SEL_NR_MENU
MOV          R1, B_OK
MOV          [R1], R0          ; Reset do B_OK
RET
```

```
ChoiceAtum      :
MOV          R1, 20
CMP          R0, R1
JNE          ChoicePorco
MOV          R7, R1
```

; Reset

```
MOV          R0, 0
MOV          R1, SEL_NR_MENU
MOV          [R1], R0          ; Reset do SEL_NR_MENU
MOV          R1, B_OK
MOV          [R1], R0          ; Reset do B_OK
RET
```

```
ChoicePorco      :
```

```
MOV            R1, 21
CMP            R0, R1
JNE            ChoiceFrango
MOV            R7, R1

; Reset
MOV            R0, 0
MOV            R1, SEL_NR_MENU
MOV            [R1], R0                ; Reset do SEL_NR_MENU
MOV            R1, B_OK
MOV            [R1], R0                ; Reset do B_OK
RET
```

```
ChoiceFrango   :
MOV            R1, 22
CMP            R0, R1
JNE            ChoicePeru
MOV            R7, R1

; Reset
MOV            R0, 0
MOV            R1, SEL_NR_MENU
MOV            [R1], R0                ; Reset do SEL_NR_MENU
MOV            R1, B_OK
MOV            [R1], R0                ; Reset do B_OK
```

RET

```
ChoicePeru          :  
MOV                 R1, 23  
CMP                 R0, R1  
JNE                 ChoiceOvo  
MOV                 R7, R1  
  
; Reset  
MOV                 R0, 0  
MOV                 R1, SEL_NR_MENU  
MOV                 [R1], R0           ; Reset do SEL_NR_MENU  
MOV                 R1, B_OK  
MOV                 [R1], R0           ; Reset do B_OK  
RET
```

```
ChoiceOvo           :  
MOV                 R1, 24  
CMP                 R0, R1  
JNE                 ChoiceQueijo  
MOV                 R7, R1  
  
; Reset  
MOV                 R0, 0
```

```
MOV          R1, SEL_NR_MENU
MOV          [R1], R0          ; Reset do SEL_NR_MENU
MOV          R1, B_OK
MOV          [R1], R0          ; Reset do B_OK
RET
```

```
ChoiceQueijo      :
MOV          R1, 25
CMP          R0, R1
JNE          MenuChangeFoodChoiceError
MOV          R7, R1

; Reset
MOV          R0, 0
MOV          R1, SEL_NR_MENU
MOV          [R1], R0          ; Reset do SEL_NR_MENU
MOV          R1, B_OK
MOV          [R1], R0          ; Reset do B_OK
RET
```

```
MenuChangeFoodChoiceError :
MOV          R0, 0
```



```

MOV          R1, SEL_NR_MENU
MOV          [R1], R0          ; Reset do SEL_NR_MENU
MOV          R1, B_OK
MOV          [R1], R0          ; Reset do B_OK
Call         MenuChoiceErrorCall

```

MenuScaleCall: ; ((), (), (), (), (), (), PesoAnterior, AlimentoAtual)

; R4 guarda o número caracteres da tabela de cada alimento, para ser usado como padding

; R5 guarda cópia do AlimentoAtual

; R8 é usado numa chamada dentro desta

; Display Scale Menu

```

CALL         PrepareDisplayCall          ; Preparar ecrã para
mostrar o menu balança

```

```

MOV          R2, GUIMenuScale            ; Guardar em R2 o
endereço do menu balança

```

```

CALL         DisplayMenuCall             ; Mostrar Menu balança
no Display

```

; Display Overwrite Peso

```

MOV          R0, PESO                    ; Move endereço de PESO
para o R0

```

```

MOV          R1, [R0]                    ; Move valor de PESO para o
R1

```

```

MOV          R2, 2500

```

```

SUB          R1, R2

```

```

JN           MenuScaleNoWeightOverflow

```

; Se valor do peso excede 2500 gramas

```

MOV          R0, PESO

```

```
MOV                R1, 0000H

MOV                [R0], R1                ; Reset do valor do peso para
0

; Se o valor do peso não excede 2500 gramas

MenuScaleNoWeightOverflow      :

MOV                R0, PESO

CALL               ConvertMemoryToASCII    ; Converte o valor do
PESO em ASCII

MOV                R0, DisplayBeginning

MOV                R1, 1                  ; Linha a dar overwrite, sendo a
primeira a linha 0

MOV                R2, DisplayNumber0      ; Endereço com o
conteúdo que irá substituir a linha

MOV                R3, 0                  ; Numero de bytes que já
levaram overwrite

CALL               OverwriteDisplayFourBytesCall

; Display Overwrite Alimento

MOV                R0, DisplayBeginning

MOV                R1, 4                  ; Linha a dar overwrite, sendo a
primeira a linha 0

MOV                R2, TableAveia         ; Endereço com o
conteúdo que irá substituir a linha

MOV                R3, 0                  ; Numero de bytes que já
levaram overwrite

MOV                R4, 32                  ; Guarda em R4 o número de
carateres da tabela de cada alimento, para ser usado como padding

MOV                R5, R7                  ; Guarda cópia do
AlimentoAtual em R5
```

MUL R5, R4 ; Constrói o padding que será usado para aceder ao endereço correto (Cópia AlimentoAtual * Nº Carateres Alimento)

ADD R2, R5 ; Adiciona o padding ao endereço da primeira tabela, obtendo a tabela que se quer aceder

CALL OverwriteDisplayCall

MenuScaleDisplayReady :

; Verificar se o botão B_OK foi pressionado

MOV R0, B_OK

MOV R1, [R0] ; Guardar valor B_OK no R1

CMP R1, 1 ; Verificar se B_OK foi pressionado

JNE MenuScaleIntermediaryOkNotPressed

; Se o botão B_OK foi pressionado

MOV R3, PESO

MOV R3, [R3] ; Mover valor do PESO para R3

MenuScaleAveia :

CMP R7, 0

JNE MenuScalePaoDeForma

MOV R0, 11

MOV em R1	R1, R3	; Cria cópia do valor do PESO
Call da macro em R0	RoundMacros	; Guarda valor arredonado
MOV	R1, PROTEINA	
MOV guardada em memória para R1	R1, [R1]	; Move valor da proteína
ADD atual à guardada em memória	R0, R1	; Adiciona valor da proteína
MOV	R1, PROTEINA	
MOV para a memória	[R1], R0	; Move novo valor da proteína
MOV	R0, 56	
MOV em R1	R1, R3	; Cria cópia do valor do PESO
Call da macro em R0	RoundMacros	; Guarda valor arredonado
MOV	R1, HIDRATOS	
MOV guardados em memória para R1	R1, [R1]	; Move valor dos hidratos
ADD atuais à guardada em memória	R0, R1	; Adiciona valor dos hidratos
MOV	R1, HIDRATOS	
MOV hidratos para a memória	[R1], R0	; Move novo valor dos
MOV	R0, 7	
MOV em R1	R1, R3	; Cria cópia do valor do PESO
Call da macro em R0	RoundMacros	; Guarda valor arredonado
MOV	R1, GORDURA	

```
MOV          R1, [R1]          ; Move valor da GORDURA
guardada em memória para R1

ADD          R0, R1            ; Adiciona valor da GORDURA
atual à guardada em memória

MOV          R1, GORDURA

MOV          [R1], R0          ; Move novo valor da
GORDURA para a memória

CALL         CalculateCalories ; Calcula o valor das
calorias

MOV          R0, B_OK

MOV          R1, 0

MOV          [R0], R1          ; Reset do botão B_OK

RET
```

```
MenuScaleIntermediaryOkNotPressed :
JMP          MenuScaleOkNotPressed
```

```

MenuScalePaoDeForma      :

CMP                        R7, 1

JNE                        MenuScaleBatata


MOV                        R0, 9

MOV                        R1, R3                ; Cria cópia do valor do PESO
em R1

Call                      RoundMacros          ; Guarda valor arredondado
da macro em R0

MOV                        R1, PROTEINA

MOV                        R1, [R1]              ; Move valor da proteína
guardada em memória para R1

ADD                        R0, R1                ; Adiciona valor da proteína
atual à guardada em memória

MOV                        R1, PROTEINA

MOV                        [R1], R0             ; Move novo valor da proteína
para a memória


MOV                        R0, 42

MOV                        R1, R3                ; Cria cópia do valor do PESO
em R1

Call                      RoundMacros          ; Guarda valor arredondado
da macro em R0

MOV                        R1, HIDRATOS

MOV                        R1, [R1]              ; Move valor dos hidratos
guardados em memória para R1

ADD                        R0, R1                ; Adiciona valor dos hidratos
atuais à guardada em memória

MOV                        R1, HIDRATOS

MOV                        [R1], R0             ; Move novo valor dos
hidratos para a memória

```

MOV	R0, 3	
MOV em R1	R1, R3	; Cria cópia do valor do PESO
Call da macro em R0	RoundMacros	; Guarda valor arredonado
MOV	R1, GORDURA	
MOV guardada em memória para R1	R1, [R1]	; Move valor da GORDURA
ADD atual à guardada em memoria	R0, R1	; Adiciona valor da GORDURA
MOV	R1, GORDURA	
MOV GORDURA para a memória	[R1], R0	; Move novo valor da
CALL calorias	CalculateCalories	; Calcula o valor das
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

```

MenuScaleBatata          :
CMP                      R7, 2
JNE                      MenuScaleArroz

MOV                      R0, 3

MOV                      R1, R3                ; Cria cópia do valor do PESO
em R1

Call                    RoundMacros            ; Guarda valor arredondado
da macro em R0

MOV                      R1, PROTEINA

MOV                      R1, [R1]              ; Move valor da proteína
guardada em memória para R1

ADD                      R0, R1                ; Adiciona valor da proteína
atual à guardada em memória

MOV                      R1, PROTEINA

MOV                      [R1], R0             ; Move novo valor da proteína
para a memória


MOV                      R0, 19

MOV                      R1, R3                ; Cria cópia do valor do PESO
em R1

Call                    RoundMacros            ; Guarda valor arredondado
da macro em R0

MOV                      R1, HIDRATOS

MOV                      R1, [R1]              ; Move valor dos hidratos
guardados em memória para R1

ADD                      R0, R1                ; Adiciona valor dos hidratos
atuais à guardada em memória

MOV                      R1, HIDRATOS

MOV                      [R1], R0             ; Move novo valor dos
hidratos para a memória

```


MOV	R0, 0	
MOV em R1	R1, R3	; Cria cópia do valor do PESO
Call da macro em R0	RoundMacros	; Guarda valor arredondado
MOV	R1, GORDURA	
MOV guardada em memória para R1	R1, [R1]	; Move valor da GORDURA
ADD atual à guardada em memória	R0, R1	; Adiciona valor da GORDURA
MOV	R1, GORDURA	
MOV GORDURA para a memória	[R1], R0	; Move novo valor da
CALL calorias	CalculateCalories	; Calcula o valor das
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

```

MenuScaleArroz          :
CMP                      R7, 3
JNE                      MenuScaleFeijao

MOV                      R0, 7

MOV                      R1, R3          ; Cria cópia do valor do PESO
em R1

Call                    RoundMacros      ; Guarda valor arredonado
da macro em R0

MOV                      R1, PROTEINA

MOV                      R1, [R1]        ; Move valor da proteina
guardada em memória para R1

ADD                      R0, R1          ; Adiciona valor da proteina
atual à guardada em memoria

MOV                      R1, PROTEINA

MOV                      [R1], R0        ; Move novo valor da proteína
para a memória


MOV                      R0, 25

MOV                      R1, R3          ; Cria cópia do valor do PESO
em R1

Call                    RoundMacros      ; Guarda valor arredonado
da macro em R0

MOV                      R1, HIDRATOS

MOV                      R1, [R1]        ; Move valor dos hidratos
guardados em memória para R1

ADD                      R0, R1          ; Adiciona valor dos hidratos
atuais à guardada em memoria

MOV                      R1, HIDRATOS

MOV                      [R1], R0        ; Move novo valor dos
hidratos para a memória


MOV                      R0, 0
    
```

MOV em R1	R1, R3	; Cria cópia do valor do PESO
Call da macro em R0	RoundMacros	; Guarda valor arredonado
MOV	R1, GORDURA	
MOV guardada em memória para R1	R1, [R1]	; Move valor da GORDURA
ADD atual à guardada em memoria	R0, R1	; Adiciona valor da GORDURA
MOV	R1, GORDURA	
MOV GORDURA para a memória	[R1], R0	; Move novo valor da
CALL calorias	CalculateCalories	; Calcula o valor das
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

```

MenuScaleFeijao          :
CMP                        R7, 4
JNE                        MenuScaleLegumes

MOV                        R0, 10

MOV                        R1, R3                ; Cria cópia do valor do PESO
em R1

Call                       RoundMacros          ; Guarda valor arredonado
da macro em R0

MOV                        R1, PROTEINA

MOV                        R1, [R1]              ; Move valor da proteina
guardada em memória para R1

ADD                        R0, R1                ; Adiciona valor da proteina
atual à guardada em memoria

MOV                        R1, PROTEINA

MOV                        [R1], R0              ; Move novo valor da proteína
para a memória

MOV                        R0, 13

MOV                        R1, R3                ; Cria cópia do valor do PESO
em R1

Call                       RoundMacros          ; Guarda valor arredonado
da macro em R0

MOV                        R1, HIDRATOS

MOV                        R1, [R1]              ; Move valor dos hidratos
guardados em memória para R1

ADD                        R0, R1                ; Adiciona valor dos hidratos
atuais à guardada em memoria

MOV                        R1, HIDRATOS

MOV                        [R1], R0              ; Move novo valor dos
hidratos para a memória

```

```
MOV                R0, 0

MOV                R1, R3                ; Cria cópia do valor do PESO
em R1

Call               RoundMacros           ; Guarda valor arredonado
da macro em R0

MOV                R1, GORDURA

MOV                R1, [R1]              ; Move valor da GORDURA
guardada em memória para R1

ADD                R0, R1                ; Adiciona valor da GORDURA
atual à guardada em memória

MOV                R1, GORDURA

MOV                [R1], R0              ; Move novo valor da
GORDURA para a memória

CALL              CalculateCalories      ; Calcula o valor das
calorias

MOV                R0, B_OK

MOV                R1, 0

MOV                [R0], R1              ; Reset do botão B_OK

RET
```

```

MenuScaleLegumes          :
CMP                        R7, 5
JNE                        MenuScaleTomate

MOV                        R0, 3
MOV                        R1, R3                ; Cria cópia do valor do PESO
em R1

Call                      RoundMacros          ; Guarda valor arredonado
da macro em R0

MOV                        R1, PROTEINA
MOV                        R1, [R1]            ; Move valor da proteina
guardada em memória para R1

ADD                        R0, R1              ; Adiciona valor da proteina
atual à guardada em memoria

MOV                        R1, PROTEINA
MOV                        [R1], R0           ; Move novo valor da proteína
para a memória

MOV                        R0, 7
MOV                        R1, R3                ; Cria cópia do valor do PESO
em R1

Call                      RoundMacros          ; Guarda valor arredonado
da macro em R0

MOV                        R1, HIDRATOS
MOV                        R1, [R1]            ; Move valor dos hidratos
guardados em memória para R1

ADD                        R0, R1              ; Adiciona valor dos hidratos
atuais à guardada em memoria

```

MOV	R1, HIDRATOS	
MOV	[R1], R0	; Move novo valor dos
hidratos para a memória		
MOV	R0, 0	
MOV	R1, R3	; Cria cópia do valor do PESO
em R1		
Call	RoundMacros	; Guarda valor arredonado
da macro em R0		
MOV	R1, GORDURA	
MOV	R1, [R1]	; Move valor da GORDURA
guardada em memória para R1		
ADD	R0, R1	; Adiciona valor da GORDURA
atual à guardada em memoria		
MOV	R1, GORDURA	
MOV	[R1], R0	; Move novo valor da
GORDURA para a memória		
CALL	CalculateCalories	; Calcula o valor das
calorias		
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

```

MenuScaleTomate      :
CMP                  R7, 6
JNE                  MenuScaleBanana

MOV                  R0, 1

MOV                  R1, R3                ; Cria cópia do valor do PESO
em R1

Call                 RoundMacros          ; Guarda valor arredonado
da macro em R0

MOV                  R1, PROTEINA

MOV                  R1, [R1]              ; Move valor da proteina
guardada em memória para R1

ADD                  R0, R1                ; Adiciona valor da proteina
atual à guardada em memoria

MOV                  R1, PROTEINA

MOV                  [R1], R0              ; Move novo valor da proteína
para a memória


MOV                  R0, 3

MOV                  R1, R3                ; Cria cópia do valor do PESO
em R1

Call                 RoundMacros          ; Guarda valor arredonado
da macro em R0

MOV                  R1, HIDRATOS

```


MOV	R1, [R1]	; Move valor dos hidratos guardados em memória para R1
ADD	R0, R1	; Adiciona valor dos hidratos atuais à guardada em memória
MOV	R1, HIDRATOS	
MOV	[R1], R0	; Move novo valor dos hidratos para a memória
MOV	R0, 0	
MOV	R1, R3	; Cria cópia do valor do PESO em R1
Call	RoundMacros	; Guarda valor arredondado da macro em R0
MOV	R1, GORDURA	
MOV	R1, [R1]	; Move valor da GORDURA guardada em memória para R1
ADD	R0, R1	; Adiciona valor da GORDURA atual à guardada em memória
MOV	R1, GORDURA	
MOV	[R1], R0	; Move novo valor da GORDURA para a memória
CALL	CalculateCalories	; Calcula o valor das calorias
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

```
MenuScaleBanana      :  
MOV                  R0, 7  
CMP                  R7, R0  
JNE                  MenuScaleLaranja  
  
MOV                  R0, 1  
MOV                  R1, R3                ; Cria cópia do valor do PESO  
em R1  
Call                 RoundMacros          ; Guarda valor arredondado  
da macro em R0  
MOV                  R1, PROTEINA  
MOV                  R1, [R1]              ; Move valor da proteína  
guardada em memória para R1  
ADD                  R0, R1                ; Adiciona valor da proteína  
atual à guardada em memória  
MOV                  R1, PROTEINA  
MOV                  [R1], R0              ; Move novo valor da proteína  
para a memória  
  
MOV                  R0, 23
```

MOV em R1	R1, R3	; Cria cópia do valor do PESO
Call da macro em R0	RoundMacros	; Guarda valor arredonado
MOV	R1, HIDRATOS	
MOV guardados em memória para R1	R1, [R1]	; Move valor dos hidratos
ADD atuais à guardada em memoria	R0, R1	; Adiciona valor dos hidratos
MOV	R1, HIDRATOS	
MOV hidratos para a memória	[R1], R0	; Move novo valor dos
MOV	R0, 0	
MOV em R1	R1, R3	; Cria cópia do valor do PESO
Call da macro em R0	RoundMacros	; Guarda valor arredonado
MOV	R1, GORDURA	
MOV guardada em memória para R1	R1, [R1]	; Move valor da GORDURA
ADD atual à guardada em memoria	R0, R1	; Adiciona valor da GORDURA
MOV	R1, GORDURA	
MOV GORDURA para a memória	[R1], R0	; Move novo valor da
CALL calorias	CalculateCalories	; Calcula o valor das
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK

RET

MenuScaleLaranja :

MOV R0, 8

CMP R7, R0

JNE MenuScaleMaca

MOV R0, 1

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredondado
da macro em R0

MOV R1, PROTEINA

MOV R1, [R1] ; Move valor da proteína
guardada em memória para R1

ADD	R0, R1	; Adiciona valor da proteína
atual à guardada em memoria		
MOV	R1, PROTEINA	
MOV	[R1], R0	; Move novo valor da proteína
para a memória		
MOV	R0, 12	
MOV	R1, R3	; Cria cópia do valor do PESO
em R1		
Call	RoundMacros	; Guarda valor arredonado
da macro em R0		
MOV	R1, HIDRATOS	
MOV	R1, [R1]	; Move valor dos hidratos
guardados em memória para R1		
ADD	R0, R1	; Adiciona valor dos hidratos
atuais à guardada em memoria		
MOV	R1, HIDRATOS	
MOV	[R1], R0	; Move novo valor dos
hidratos para a memória		
MOV	R0, 0	
MOV	R1, R3	; Cria cópia do valor do PESO
em R1		
Call	RoundMacros	; Guarda valor arredonado
da macro em R0		
MOV	R1, GORDURA	
MOV	R1, [R1]	; Move valor da GORDURA
guardada em memória para R1		
ADD	R0, R1	; Adiciona valor da GORDURA
atual à guardada em memoria		
MOV	R1, GORDURA	
MOV	[R1], R0	; Move novo valor da
GORDURA para a memória		

CALL CalculateCalories ; Calcula o valor das
calorias

MOV R0, B_OK

MOV R1, 0

MOV [R0], R1 ; Reset do botão B_OK

RET

MenuScaleMaca :

MOV R0, 9

CMP R7, R0

JNE MenuScaleKiwi

MOV R0, 1

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredondado
da macro em R0

MOV R1, PROTEINA

MOV R1, [R1] ; Move valor da proteína
guardada em memória para R1

ADD R0, R1 ; Adiciona valor da proteína
atual à guardada em memória

MOV R1, PROTEINA

MOV [R1], R0 ; Move novo valor da proteína
para a memória

MOV R0, 14

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredondado
da macro em R0

MOV R1, HIDRATOS

MOV R1, [R1] ; Move valor dos hidratos
guardados em memória para R1

ADD R0, R1 ; Adiciona valor dos hidratos
atuais à guardada em memória

MOV R1, HIDRATOS

MOV [R1], R0 ; Move novo valor dos
hidratos para a memória

MOV	R0, 0	
MOV em R1	R1, R3	; Cria cópia do valor do PESO
Call da macro em R0	RoundMacros	; Guarda valor arredonado
MOV	R1, GORDURA	
MOV guardada em memória para R1	R1, [R1]	; Move valor da GORDURA
ADD atual à guardada em memoria	R0, R1	; Adiciona valor da GORDURA
MOV	R1, GORDURA	
MOV GORDURA para a memória	[R1], R0	; Move novo valor da
CALL calorias	CalculateCalories	; Calcula o valor das
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		


```

MenuScaleKiwi      :
MOV                R0, 10
CMP                R7, R0
JNE                MenuScaleBolachaChoc

MOV                R0, 1
MOV                R1, R3                ; Cria cópia do valor do PESO
em R1

Call              RoundMacros            ; Guarda valor arredonado
da macro em R0

MOV                R1, PROTEINA
MOV                R1, [R1]              ; Move valor da proteína
guardada em memória para R1

ADD                R0, R1                ; Adiciona valor da proteína
atual à guardada em memória

MOV                R1, PROTEINA
MOV                [R1], R0              ; Move novo valor da proteína
para a memória

MOV                R0, 15
MOV                R1, R3                ; Cria cópia do valor do PESO
em R1

```

Call da macro em R0	RoundMacros	; Guarda valor arredonado
MOV	R1, HIDRATOS	
MOV guardados em memória para R1	R1, [R1]	; Move valor dos hidratos
ADD atuais à guardada em memoria	R0, R1	; Adiciona valor dos hidratos
MOV	R1, HIDRATOS	
MOV hidratos para a memória	[R1], R0	; Move novo valor dos
MOV	R0, 0	
MOV em R1	R1, R3	; Cria cópia do valor do PESO
Call da macro em R0	RoundMacros	; Guarda valor arredonado
MOV	R1, GORDURA	
MOV guardada em memória para R1	R1, [R1]	; Move valor da GORDURA
ADD atual à guardada em memoria	R0, R1	; Adiciona valor da GORDURA
MOV	R1, GORDURA	
MOV GORDURA para a memória	[R1], R0	; Move novo valor da
CALL calorias	CalculateCalories	; Calcula o valor das
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

MenuScaleBolachaChoc :

MOV R0, 11

CMP R7, R0

JNE MenuScalePizza

MOV R0, 9

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredondado
da macro em R0

MOV R1, PROTEINA

MOV R1, [R1] ; Move valor da proteína guardada em memória para R1

ADD R0, R1 ; Adiciona valor da proteína atual à guardada em memória

MOV R1, PROTEINA

MOV [R1], R0 ; Move novo valor da proteína para a memória

MOV R0, 59

MOV R1, R3 ; Cria cópia do valor do PESO em R1

Call RoundMacros ; Guarda valor arredondado da macro em R0

MOV R1, HIDRATOS

MOV R1, [R1] ; Move valor dos hidratos guardados em memória para R1

ADD R0, R1 ; Adiciona valor dos hidratos atuais à guardada em memória

MOV R1, HIDRATOS

MOV [R1], R0 ; Move novo valor dos hidratos para a memória

MOV R0, 22

MOV R1, R3 ; Cria cópia do valor do PESO em R1

Call RoundMacros ; Guarda valor arredondado da macro em R0

MOV R1, GORDURA

MOV R1, [R1] ; Move valor da GORDURA guardada em memória para R1

ADD R0, R1 ; Adiciona valor da GORDURA atual à guardada em memória

MOV R1, GORDURA

MOV	[R1], R0	; Move novo valor da
GORDURA para a memória		
CALL	CalculateCalories	; Calcula o valor das
calorias		
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

```

MenuScalePizza          :
MOV                      R0, 12
CMP                      R7, R0
JNE                      MenuScaleAmendoas

MOV                      R0, 13

MOV                      R1, R3          ; Cria cópia do valor do PESO
em R1

Call                    RoundMacros      ; Guarda valor arredonado
da macro em R0

MOV                      R1, PROTEINA

MOV                      R1, [R1]        ; Move valor da proteína
guardada em memória para R1

ADD                      R0, R1          ; Adiciona valor da proteína
atual à guardada em memória

MOV                      R1, PROTEINA

MOV                      [R1], R0       ; Move novo valor da proteína
para a memória


MOV                      R0, 25

MOV                      R1, R3          ; Cria cópia do valor do PESO
em R1

Call                    RoundMacros      ; Guarda valor arredonado
da macro em R0

MOV                      R1, HIDRATOS

MOV                      R1, [R1]        ; Move valor dos hidratos
guardados em memória para R1

ADD                      R0, R1          ; Adiciona valor dos hidratos
atuais à guardada em memória

MOV                      R1, HIDRATOS

MOV                      [R1], R0       ; Move novo valor dos
hidratos para a memória

```

MOV	R0, 9	
MOV em R1	R1, R3	; Cria cópia do valor do PESO
Call da macro em R0	RoundMacros	; Guarda valor arredondado
MOV	R1, GORDURA	
MOV guardada em memória para R1	R1, [R1]	; Move valor da GORDURA
ADD atual à guardada em memória	R0, R1	; Adiciona valor da GORDURA
MOV	R1, GORDURA	
MOV GORDURA para a memória	[R1], R0	; Move novo valor da
CALL calorias	CalculateCalories	; Calcula o valor das
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

MenuScaleAmendoas :

MOV R0, 13

CMP R7, R0

JNE MenuScaleLinhaca

MOV R0, 25

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredondado
da macro em R0

MOV R1, PROTEINA

MOV R1, [R1] ; Move valor da proteína
guardada em memória para R1

ADD R0, R1 ; Adiciona valor da proteína
atual à guardada em memória

MOV R1, PROTEINA

MOV [R1], R0 ; Move novo valor da proteína
para a memória

MOV R0, 6

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredondado
da macro em R0

MOV	R1, HIDRATOS	
MOV	R1, [R1]	; Move valor dos hidratos guardados em memória para R1
ADD	R0, R1	; Adiciona valor dos hidratos atuais à guardada em memória
MOV	R1, HIDRATOS	
MOV	[R1], R0	; Move novo valor dos hidratos para a memória
MOV	R0, 55	
MOV	R1, R3	; Cria cópia do valor do PESO em R1
Call	RoundMacros	; Guarda valor arredondado da macro em R0
MOV	R1, GORDURA	
MOV	R1, [R1]	; Move valor da GORDURA guardada em memória para R1
ADD	R0, R1	; Adiciona valor da GORDURA atual à guardada em memória
MOV	R1, GORDURA	
MOV	[R1], R0	; Move novo valor da GORDURA para a memória
CALL	CalculateCalories	; Calcula o valor das calorias
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

```

MenuScaleLinhaca          :
MOV                        R0, 14
CMP                        R7, R0
JNE                        MenuScaleAzeite

MOV                        R0, 18

MOV                        R1, R3                ; Cria cópia do valor do PESO
em R1

Call                       RoundMacros          ; Guarda valor arredonado
da macro em R0

MOV                        R1, PROTEINA

MOV                        R1, [R1]              ; Move valor da proteina
guardada em memória para R1

ADD                        R0, R1                ; Adiciona valor da proteina
atual à guardada em memoria

MOV                        R1, PROTEINA

MOV                        [R1], R0              ; Move novo valor da proteína
para a memória
    
```

MOV	R0, 34	
MOV em R1	R1, R3	; Cria cópia do valor do PESO
Call da macro em R0	RoundMacros	; Guarda valor arredonado
MOV	R1, HIDRATOS	
MOV guardados em memória para R1	R1, [R1]	; Move valor dos hidratos
ADD atuais à guardada em memoria	R0, R1	; Adiciona valor dos hidratos
MOV	R1, HIDRATOS	
MOV hidratos para a memória	[R1], R0	; Move novo valor dos
MOV	R0, 36	
MOV em R1	R1, R3	; Cria cópia do valor do PESO
Call da macro em R0	RoundMacros	; Guarda valor arredonado
MOV	R1, GORDURA	
MOV guardada em memória para R1	R1, [R1]	; Move valor da GORDURA
ADD atual à guardada em memoria	R0, R1	; Adiciona valor da GORDURA
MOV	R1, GORDURA	
MOV GORDURA para a memória	[R1], R0	; Move novo valor da
CALL calorias	CalculateCalories	; Calcula o valor das
MOV	R0, B_OK	
MOV	R1, 0	

MOV [R0], R1 ; Reset do botão B_OK

RET

MenuScaleAzeite :

MOV R0, 15

CMP R7, R0

JNE MenuScaleLeite

MOV R0, 0

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredonado
da macro em R0

MOV R1, PROTEINA

MOV R1, [R1] ; Move valor da proteína
guardada em memória para R1

ADD R0, R1 ; Adiciona valor da proteína
atual à guardada em memória

MOV R1, PROTEINA

MOV [R1], R0 ; Move novo valor da proteína
para a memória

MOV R0, 0

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredonado
da macro em R0

MOV	R1, HIDRATOS	
MOV	R1, [R1]	; Move valor dos hidratos guardados em memória para R1
ADD	R0, R1	; Adiciona valor dos hidratos atuais à guardada em memoria
MOV	R1, HIDRATOS	
MOV	[R1], R0	; Move novo valor dos hidratos para a memória
MOV	R0, 100	
MOV	R1, R3	; Cria cópia do valor do PESO em R1
Call	RoundMacros	; Guarda valor arredonado da macro em R0
MOV	R1, GORDURA	
MOV	R1, [R1]	; Move valor da GORDURA guardada em memória para R1
ADD	R0, R1	; Adiciona valor da GORDURA atual à guardada em memoria
MOV	R1, GORDURA	
MOV	[R1], R0	; Move novo valor da GORDURA para a memória
CALL	CalculateCalories	; Calcula o valor das calorias
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

```

MenuScaleLeite          :
MOV                      R0, 16
CMP                      R7, R0
JNE                      MenuScaleWhey

MOV                      R0, 3
MOV                      R1, R3                ; Cria cópia do valor do PESO
em R1

Call                    RoundMacros            ; Guarda valor arredonado
da macro em R0

MOV                      R1, PROTEINA
MOV                      R1, [R1]              ; Move valor da proteina
guardada em memória para R1

ADD                      R0, R1                ; Adiciona valor da proteina
atual à guardada em memoria

MOV                      R1, PROTEINA
MOV                      [R1], R0              ; Move novo valor da proteína
para a memória

MOV                      R0, 4
MOV                      R1, R3                ; Cria cópia do valor do PESO
em R1

Call                    RoundMacros            ; Guarda valor arredonado
da macro em R0

MOV                      R1, HIDRATOS
MOV                      R1, [R1]              ; Move valor dos hidratos
guardados em memória para R1

```

ADD	R0, R1	; Adiciona valor dos hidratos
atuais à guardada em memoria		
MOV	R1, HIDRATOS	
MOV	[R1], R0	; Move novo valor dos
hidratos para a memória		
MOV	R0, 0	
MOV	R1, R3	; Cria cópia do valor do PESO
em R1		
Call	RoundMacros	; Guarda valor arredonado
da macro em R0		
MOV	R1, GORDURA	
MOV	R1, [R1]	; Move valor da GORDURA
guardada em memória para R1		
ADD	R0, R1	; Adiciona valor da GORDURA
atual à guardada em memoria		
MOV	R1, GORDURA	
MOV	[R1], R0	; Move novo valor da
GORDURA para a memória		
CALL	CalculateCalories	; Calcula o valor das
calorias		
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

```

MenuScaleWhey          :
MOV                     R0, 17
CMP                     R7, R0
JNE                     MenuScaleSalmao

MOV                     R0, 80

MOV                     R1, R3                ; Cria cópia do valor do PESO
em R1

Call                    RoundMacros          ; Guarda valor arredonado
da macro em R0

MOV                     R1, PROTEINA

MOV                     R1, [R1]              ; Move valor da proteina
guardada em memória para R1

ADD                     R0, R1                ; Adiciona valor da proteina
atual à guardada em memoria

MOV                     R1, PROTEINA

MOV                     [R1], R0              ; Move novo valor da proteína
para a memória


MOV                     R0, 8

MOV                     R1, R3                ; Cria cópia do valor do PESO
em R1

Call                    RoundMacros          ; Guarda valor arredonado
da macro em R0

MOV                     R1, HIDRATOS

MOV                     R1, [R1]              ; Move valor dos hidratos
guardados em memória para R1

ADD                     R0, R1                ; Adiciona valor dos hidratos
atuais à guardada em memoria

MOV                     R1, HIDRATOS

```


MOV	[R1], R0	; Move novo valor dos
hidratos para a memória		
MOV	R0, 4	
MOV	R1, R3	; Cria cópia do valor do PESO
em R1		
Call	RoundMacros	; Guarda valor arredondado
da macro em R0		
MOV	R1, GORDURA	
MOV	R1, [R1]	; Move valor da GORDURA
guardada em memória para R1		
ADD	R0, R1	; Adiciona valor da GORDURA
atual à guardada em memória		
MOV	R1, GORDURA	
MOV	[R1], R0	; Move novo valor da
GORDURA para a memória		
CALL	CalculateCalories	; Calcula o valor das
calorias		
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

MenuScaleSalmao :

MOV R0, 18

CMP R7, R0

JNE MenuScalePescada

MOV R0, 21

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredonado
da macro em R0

MOV R1, PROTEINA

MOV R1, [R1] ; Move valor da proteina
guardada em memória para R1

ADD R0, R1 ; Adiciona valor da proteina
atual à guardada em memoria

MOV R1, PROTEINA

MOV [R1], R0 ; Move novo valor da proteína
para a memória

MOV R0, 0

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredonado
da macro em R0

MOV	R1, HIDRATOS	
MOV	R1, [R1]	; Move valor dos hidratos guardados em memória para R1
ADD	R0, R1	; Adiciona valor dos hidratos atuais à guardada em memoria
MOV	R1, HIDRATOS	
MOV	[R1], R0	; Move novo valor dos hidratos para a memória
MOV	R0, 15	
MOV	R1, R3	; Cria cópia do valor do PESO em R1
Call	RoundMacros	; Guarda valor arredonado da macro em R0
MOV	R1, GORDURA	
MOV	R1, [R1]	; Move valor da GORDURA guardada em memória para R1
ADD	R0, R1	; Adiciona valor da GORDURA atual à guardada em memoria
MOV	R1, GORDURA	
MOV	[R1], R0	; Move novo valor da GORDURA para a memória
CALL	CalculateCalories	; Calcula o valor das calorias
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

MenuScalePescada :

MOV R0, 19

CMP R7, R0

JNE MenuScaleAtum

MOV R0, 20

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredondado
da macro em R0

MOV R1, PROTEINA

MOV R1, [R1] ; Move valor da proteína
guardada em memória para R1

ADD R0, R1 ; Adiciona valor da proteína
atual à guardada em memória

MOV R1, PROTEINA

MOV [R1], R0 ; Move novo valor da proteína
para a memória

MOV R0, 0

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredondado
da macro em R0

MOV	R1, HIDRATOS	
MOV	R1, [R1]	; Move valor dos hidratos guardados em memória para R1
ADD	R0, R1	; Adiciona valor dos hidratos atuais à guardada em memoria
MOV	R1, HIDRATOS	
MOV	[R1], R0	; Move novo valor dos hidratos para a memória
MOV	R0, 1	
MOV	R1, R3	; Cria cópia do valor do PESO em R1
Call	RoundMacros	; Guarda valor arredonado da macro em R0
MOV	R1, GORDURA	
MOV	R1, [R1]	; Move valor da GORDURA guardada em memória para R1
ADD	R0, R1	; Adiciona valor da GORDURA atual à guardada em memoria
MOV	R1, GORDURA	
MOV	[R1], R0	; Move novo valor da GORDURA para a memória
CALL	CalculateCalories	; Calcula o valor das calorias
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

```

MenuScaleAtum          :
MOV                    R0, 20
CMP                    R7, R0
JNE                    MenuScalePorco

MOV                    R0, 25

MOV                    R1, R3                ; Cria cópia do valor do PESO
em R1

Call                   RoundMacros           ; Guarda valor arredonado
da macro em R0

MOV                    R1, PROTEINA

MOV                    R1, [R1]              ; Move valor da proteína
guardada em memória para R1

ADD                    R0, R1                ; Adiciona valor da proteína
atual à guardada em memória

MOV                    R1, PROTEINA

MOV                    [R1], R0             ; Move novo valor da proteína
para a memória


MOV                    R0, 0

MOV                    R1, R3                ; Cria cópia do valor do PESO
em R1

Call                   RoundMacros           ; Guarda valor arredonado
da macro em R0

MOV                    R1, HIDRATOS

```

MOV	R1, [R1]	; Move valor dos hidratos guardados em memória para R1
ADD	R0, R1	; Adiciona valor dos hidratos atuais à guardada em memória
MOV	R1, HIDRATOS	
MOV	[R1], R0	; Move novo valor dos hidratos para a memória
MOV	R0, 2	
MOV	R1, R3	; Cria cópia do valor do PESO em R1
Call	RoundMacros	; Guarda valor arredondado da macro em R0
MOV	R1, GORDURA	
MOV	R1, [R1]	; Move valor da GORDURA guardada em memória para R1
ADD	R0, R1	; Adiciona valor da GORDURA atual à guardada em memória
MOV	R1, GORDURA	
MOV	[R1], R0	; Move novo valor da GORDURA para a memória
CALL	CalculateCalories	; Calcula o valor das calorias
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

```

MenuScalePorco          :
MOV                     R0, 21
CMP                     R7, R0
JNE                     MenuScaleFrango

MOV                     R0, 22

MOV                     R1, R3                ; Cria cópia do valor do PESO
em R1

Call                   RoundMacros            ; Guarda valor arredonado
da macro em R0

MOV                     R1, PROTEINA

MOV                     R1, [R1]              ; Move valor da proteína
guardada em memória para R1

ADD                     R0, R1                ; Adiciona valor da proteína
atual à guardada em memória

MOV                     R1, PROTEINA

MOV                     [R1], R0              ; Move novo valor da proteína
para a memória


MOV                     R0, 0

MOV                     R1, R3                ; Cria cópia do valor do PESO
em R1

Call                   RoundMacros            ; Guarda valor arredonado
da macro em R0

MOV                     R1, HIDRATOS

```


MOV	R1, [R1]	; Move valor dos hidratos guardados em memória para R1
ADD	R0, R1	; Adiciona valor dos hidratos atuais à guardada em memória
MOV	R1, HIDRATOS	
MOV	[R1], R0	; Move novo valor dos hidratos para a memória
MOV	R0, 15	
MOV	R1, R3	; Cria cópia do valor do PESO em R1
Call	RoundMacros	; Guarda valor arredondado da macro em R0
MOV	R1, GORDURA	
MOV	R1, [R1]	; Move valor da GORDURA guardada em memória para R1
ADD	R0, R1	; Adiciona valor da GORDURA atual à guardada em memória
MOV	R1, GORDURA	
MOV	[R1], R0	; Move novo valor da GORDURA para a memória
CALL	CalculateCalories	; Calcula o valor das calorias
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

MenuScaleFrango :

MOV R0, 22

CMP R7, R0

JNE MenuScalePeru

MOV R0, 25

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredondado
da macro em R0

MOV R1, PROTEINA

MOV R1, [R1] ; Move valor da proteína
guardada em memória para R1

ADD R0, R1 ; Adiciona valor da proteína
atual à guardada em memória

MOV R1, PROTEINA

MOV [R1], R0 ; Move novo valor da proteína
para a memória

MOV R0, 0

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredondado
da macro em R0

MOV R1, HIDRATOS

MOV R1, [R1] ; Move valor dos hidratos
guardados em memória para R1

ADD R0, R1 ; Adiciona valor dos hidratos
atuais à guardada em memória

MOV	R1, HIDRATOS	
MOV	[R1], R0	; Move novo valor dos
hidratos para a memória		
MOV	R0, 4	
MOV	R1, R3	; Cria cópia do valor do PESO
em R1		
Call	RoundMacros	; Guarda valor arredonado
da macro em R0		
MOV	R1, GORDURA	
MOV	R1, [R1]	; Move valor da GORDURA
guardada em memória para R1		
ADD	R0, R1	; Adiciona valor da GORDURA
atual à guardada em memoria		
MOV	R1, GORDURA	
MOV	[R1], R0	; Move novo valor da
GORDURA para a memória		
CALL	CalculateCalories	; Calcula o valor das
calorias		
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

```

MenuScalePeru          :
MOV                     R0, 23
CMP                     R7, R0
JNE                     MenuScaleOvo

MOV                     R0, 28
MOV                     R1, R3                ; Cria cópia do valor do PESO
em R1

Call                    RoundMacros          ; Guarda valor arredonado
da macro em R0

MOV                     R1, PROTEINA
MOV                     R1, [R1]             ; Move valor da proteína
guardada em memória para R1

ADD                     R0, R1              ; Adiciona valor da proteína
atual à guardada em memória

MOV                     R1, PROTEINA
MOV                     [R1], R0            ; Move novo valor da proteína
para a memória

MOV                     R0, 0
MOV                     R1, R3                ; Cria cópia do valor do PESO
em R1

Call                    RoundMacros          ; Guarda valor arredonado
da macro em R0

MOV                     R1, HIDRATOS
MOV                     R1, [R1]             ; Move valor dos hidratos
guardados em memória para R1

ADD                     R0, R1              ; Adiciona valor dos hidratos
atuais à guardada em memória

MOV                     R1, HIDRATOS
MOV                     [R1], R0            ; Move novo valor dos
hidratos para a memória

```

```
MOV                R0, 1

MOV                R1, R3                ; Cria cópia do valor do PESO
em R1

Call              RoundMacros            ; Guarda valor arredondado
da macro em R0

MOV                R1, GORDURA

MOV                R1, [R1]              ; Move valor da GORDURA
guardada em memória para R1

ADD                R0, R1                ; Adiciona valor da GORDURA
atual à guardada em memória

MOV                R1, GORDURA

MOV                [R1], R0              ; Move novo valor da
GORDURA para a memória

CALL              CalculateCalories      ; Calcula o valor das
calorias

MOV                R0, B_OK

MOV                R1, 0

MOV                [R0], R1              ; Reset do botão B_OK

RET
```

```

MenuScaleOvo          :
MOV                    R0, 24
CMP                    R7, R0
JNE                    MenuScaleQueijo

MOV                    R0, 7
MOV                    R1, R3          ; Cria cópia do valor do PESO
em R1

Call                   RoundMacros      ; Guarda valor arredondado
da macro em R0

MOV                    R1, PROTEINA
MOV                    R1, [R1]         ; Move valor da proteína
guardada em memória para R1

ADD                    R0, R1          ; Adiciona valor da proteína
atual à guardada em memória

MOV                    R1, PROTEINA
MOV                    [R1], R0        ; Move novo valor da proteína
para a memória

MOV                    R0, 0
MOV                    R1, R3          ; Cria cópia do valor do PESO
em R1

Call                   RoundMacros      ; Guarda valor arredondado
da macro em R0

MOV                    R1, HIDRATOS
MOV                    R1, [R1]         ; Move valor dos hidratos
guardados em memória para R1

ADD                    R0, R1          ; Adiciona valor dos hidratos
atuais à guardada em memória

MOV                    R1, HIDRATOS
MOV                    [R1], R0        ; Move novo valor dos
hidratos para a memória

```

MOV	R0, 5	
MOV em R1	R1, R3	; Cria cópia do valor do PESO
Call da macro em R0	RoundMacros	; Guarda valor arredonado
MOV	R1, GORDURA	
MOV guardada em memória para R1	R1, [R1]	; Move valor da GORDURA
ADD atual à guardada em memoria	R0, R1	; Adiciona valor da GORDURA
MOV	R1, GORDURA	
MOV GORDURA para a memória	[R1], R0	; Move novo valor da
CALL calorias	CalculateCalories	; Calcula o valor das
MOV	R0, B_OK	
MOV	R1, 0	
MOV	[R0], R1	; Reset do botão B_OK
RET		

MenuScaleQueijo :

MOV R0, 25

CMP R7, R0

JNE MenuScaleQueijo

MOV R0, 28

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredonado
da macro em R0

MOV R1, PROTEINA

MOV R1, [R1] ; Move valor da proteína
guardada em memória para R1

ADD R0, R1 ; Adiciona valor da proteína
atual à guardada em memória

MOV R1, PROTEINA

MOV [R1], R0 ; Move novo valor da proteína
para a memória

MOV R0, 0

MOV R1, R3 ; Cria cópia do valor do PESO
em R1

Call RoundMacros ; Guarda valor arredonado
da macro em R0

MOV R1, HIDRATOS

MOV R1, [R1] ; Move valor dos hidratos
guardados em memória para R1

ADD R0, R1 ; Adiciona valor dos hidratos
atuais à guardada em memória

MOV R1, HIDRATOS

MOV [R1], R0 ; Move novo valor dos
hidratos para a memória


```
MOV                                R0, 13

MOV                                R1, R3                                ; Cria cópia do valor do PESO
em R1

Call                               RoundMacros                        ; Guarda valor arredonado
da macro em R0

MOV                                R1, GORDURA

MOV                                R1, [R1]                            ; Move valor da GORDURA
guardada em memória para R1

ADD                                R0, R1                            ; Adiciona valor da GORDURA
atual à guardada em memoria

MOV                                R1, GORDURA

MOV                                [R1], R0                          ; Move novo valor da
GORDURA para a memória

CALL                               CalculateCalories                  ; Calcula o valor das
calorias

MOV                                R0, B_OK

MOV                                R1, 0

MOV                                [R0], R1                          ; Reset do botão B_OK

RET

; Se o botão B_OK não foi pressionado
MenuScaleOkNotPressed            :

; Verificar se o botão change foi pressionado

MOV                                R0, B_CHANGE

MOV                                R1, [R0]                            ; Guardar valor B_CHANGE no
R1
```

```
CMP          R1, 1          ; Verificar se B_CHANGE foi
pressionado

JNE          MenuScaleChangeNotPressed

; Se o botão change foi pressionado

MOV          R8, 0          ; Passa 0 como parâmetro para
mostrar o primeiro menu da tabela

MOV          R0, B_CHANGE

MOV          [R0], R8        ; Reset do periférico
[B_CHANGE] antes de entrar no próximo menu

Call         MenuChangeFoodCall

JMP          MenuScaleCall

; Se o botão change não foi pressionado

MenuScaleChangeNotPressed :

MOV          R0, PESO

MOV          R1, [R0]        ; Guardar valor atual do PESO
no R1

CMP          R1, R6          ; Comparar valor atual do
peso, com o valor guardado anteriormente

MOV          R6, R1          ; Criar cópia do valor atual do
peso, na próxima iteração será comparada com o novo valor do peso nessa iteração

; Se o peso mudou comparado com a iteração anterior

JNE          MenuScaleIntermediaryCall

; Se o peso não mudou comparado com a iteração anterior

JMP          MenuScaleDisplayReady

MenuScaleIntermediaryCall :

JMP          MenuScaleCall
```

MenuDailyTotalCall:

; Display

CALL PrepareDisplayCall ; Preparar Display para
mostrar o total diário

MOV R2, GUIMenuDailyTotal ; Guardar em R2 o
endereço do menu total diário

CALL DisplayMenuCall ; Mostrar Menu total
diário no Display

MOV R0, PROTEINA

CALL ConvertMemoryToASCII ; Converte o valor da
proteína em ASCII

MOV R0, DisplayBeginning

MOV R1, 1

MOV R2, DisplayNumber0

MOV R3, 0

CALL OverwriteDisplayFourBytesCall

MOV R0, HIDRATOS

CALL ConvertMemoryToASCII ; Converte o valor
dos hidratos em ASCII

MOV R0, DisplayBeginning

MOV R1, 2

MOV R2, DisplayNumber0

MOV R3, 0

CALL OverwriteDisplayFourBytesCall

MOV R0, GORDURA

CALL ConvertMemoryToASCII ; Converte o valor da
gordura em ASCII

```
MOV                R0, DisplayBeginning
MOV                R1, 3
MOV                R2, DisplayNumber0
MOV                R3, 0
CALL               OverwriteDisplayFourBytesCall
```

```
MOV                R0, CALORIAS
CALL               ConvertMemoryToASCII           ; Converte o valor das
calorias em ASCII
```

```
MOV                R0, DisplayBeginning
MOV                R1, 6
MOV                R2, DisplayNumber0
MOV                R3, 0
CALL               OverwriteDisplayFourBytesCall
```

MenuDailyTotalDisplayReady :

; Verificar se o B_OK foi pressionado

```
MOV                R0, B_OK
MOV                R0, [R0]
CMP                R0, 1
JNE                MenuDailyTotalOkNotPressed
```

; Se o B_OK foi pressionado

```
MOV                R0, 0
MOV                R1, B_OK
MOV                [R1], R0                ; Reset do B_OK
```

RET

; Se o B_OK não foi pressionado

MenuDailyTotalOkNotPressed :

JMP MenuDailyTotalDisplayReady

MenuDailyGoalSeeCall:

; Display Scale Menu

CALL PrepareDisplayCall ; Preparar ecrã para
mostrar o menu mudar meta

MOV R2, GUIMenuDailyGoalSee ; Guardar em R2 o
endereço do menu mudar meta

CALL DisplayMenuCall ; Mostrar Menu mudar
meta no Display

MOV R0, META_PROTEINA

MOV R0, [R0] ; Move valor
META_PROTEINA para R0

MOV R1, PROTEINA

MOV R1, [R1] ; Move valor PROTEINA para
R1

SUB R0, R1 ; META PROTEINA - PROTEINA

JNN MetaProteinaNaoUltrapassada

; Se a meta foi ultrapassada

MOV R0, 0 ; Dá override no registo 0,
mudando o resultado negativo da meta - macro para 0

; Se a meta não foi ultrapassada

MetaProteinaNaoUltrapassada :

MOV R1, DIFERENCA_PROTEINA

MOV [R1], R0

MOV R0, DIFERENCA_PROTEINA

CALL ConvertMemoryToASCII ; Converte o valor da
PROTEINA em ASCII

```

MOV          R0, DisplayBeginning
MOV          R1, 2                      ; Linha a dar overwrite, sendo a
primeira a linha 0
MOV          R2, DisplayNumber0        ; Endereço com o
conteúdo que irá substituir a linha
MOV          R3, 0                      ; Numero de bytes que já
levaram overwrite
CALL         OverwriteDisplayFourBytesCall

```

```

MOV          R0, META_HIDRATOS
MOV          R0, [R0]                  ; Move valor
META_HIDRATOS para R0
MOV          R1, HIDRATOS
MOV          R1, [R1]                  ; Move valor HIDRATOS para
R1
SUB          R0, R1                    ; META_HIDRATOS - HIDRATOS
JNN          MetaHidratosNaoUltrapassada
; Se a meta foi ultrapassada
MOV          R0, 0                     ; Dá override no registo 0,
mudando o resultado negativo da meta - macro para 0
; Se a meta não foi ultrapassada
MetaHidratosNaoUltrapassada          :
MOV          R1, DIFERENCA_HIDRATOS
MOV          [R1], R0

MOV          R0, DIFERENCA_HIDRATOS
CALL         ConvertMemoryToASCII      ; Converte o valor do
HIDRATOS em ASCII
MOV          R0, DisplayBeginning
MOV          R1, 3                      ; Linha a dar overwrite, sendo a
primeira a linha 0

```

MOV R2, DisplayNumber0 ; Endereço com o conteúdo que irá substituir a linha

MOV R3, 0 ; Numero de bytes que já levaram overwrite

CALL OverwriteDisplayFourBytesCall

MOV R0, META_GORDURA

MOV R0, [R0] ; Move valor META_GORDURA para R0

MOV R1, GORDURA

MOV R1, [R1] ; Move valor GORDURA para R1

SUB R0, R1 ; META_GORDURA - GORDURA

JNN MetaGorduraNaoUltrapassada

; Se a meta foi ultrapassada

MOV R0, 0 ; Dá override no registo 0, mudando o resultado negativo da meta - macro para 0

; Se a meta não foi ultrapassada

MetaGorduraNaoUltrapassada :

MOV R1, DIFERENCA_GORDURA

MOV [R1], R0

MOV R0, DIFERENCA_GORDURA

CALL ConvertMemoryToASCII ; Converte o valor do GORDURA em ASCII

MOV R0, DisplayBeginning

MOV R1, 4 ; Linha a dar overwrite, sendo a primeira a linha 0

MOV R2, DisplayNumber0 ; Endereço com o conteúdo que irá substituir a linha

```

MOV          R3, 0                ; Numero de bytes que já
levaram overwrite

CALL         OverwriteDisplayFourBytesCall


MOV          R0, META_CALORIAS

MOV          R0, [R0]              ; Move valor META_CALORIAS
para R0

MOV          R1, CALORIAS

MOV          R1, [R1]              ; Move valor CALORIAS para
R1

SUB          R0, R1                ; META_CALORIAS - CALORIAS

JNN          MetaCatoriaNaoUltrapassada

; Se a meta foi ultrapassada

MOV          R0, 0                ; Dá override no registo 0,
mudando o resultado negativo da meta - macro para 0

; Se a meta não foi ultrapassada

MetaCatoriaNaoUltrapassada      :

MOV          R1, DIFERENCA_CALORIAS

MOV          [R1], R0


MOV          R0, DIFERENCA_CALORIAS

CALL         ConvertMemoryToASCII    ; Converte o valor das
calorias em ASCII

MOV          R0, DisplayBeginning

MOV          R1, 6                ; Linha a dar overwrite, sendo a
primeira a linha 0

MOV          R2, DisplayNumber0      ; Endereço com o
conteúdo que irá substituir a linha

MOV          R3, 0                ; Numero de bytes que já
levaram overwrite

CALL         OverwriteDisplayFourBytesCall

```


MenuDailyGoalSeeDisplayReady :

MOV R0, B_OK

MOV R1, [R0] ; Escrever o valor de B_OK em R1

; Verificar se o utilizador confirmou a sua escolha

CMP R1, 1 ; Verificar se o utilizador premiu o botão de confirmar a escolha

JNE MenuDailyGoalSeeDisplayReady ; Se o utilizador não clicou confirmar a escolha ainda, voltar a verificar

; Se o utilizador confirmou a escolha

RET

MenuDailyGoalChangeCall:

; Display

CALL PrepareDisplayCall ; preparar display para mostrar o menu mudar meta

MOV R2, GUIMenuDailyGoalChange ; Guardar em R2 o endereço do menu mudar meta

CALL DisplayMenuCall

MenuDailyGoalChangeDisplayReady :

MOV R0, B_OK

MOV R1, [R0] ; Escrever o valor de B_OK em R1

; Verificar se o utilizador confirmou a sua escolha

CMP R1, 1 ; Verificar se o utilizador premiu o botão de confirmar a escolha

JNE MenuDailyGoalChangeDisplayReady ; Se o utilizador não clicou confirmar a escolha ainda, voltar a verificar

; Se o utilizador confirmou a escolha

MOV R0, SEL_NR_MENU

MOV R1, [R0] ; Escrever o valor de SEL_NR_MENU em R1

; Escolha 1 - Proteína

CMP R1, 1

JNE ChoiceDailyGoalChangeHidratos ; Se não foi esta a escolha do utilizador, verifica a próxima

; Input

MOV R1, 0

MOV R0, SEL_NR_MENU

MOV [R0], R1 ; Reset do periférico
[SEL_NR_MENU] antes de entrar no próximo menu

MOV R0, B_OK

MOV [R0], R1 ; Reset do periférico [B_OK]
antes de entrar no próximo menu

; Mudar valor da meta

MOV R0, META_PROTEINA ; Mover para R0 o valor da meta

MOV R1, PESO

MOV R1, [R1] ; Mover para R1 o valor do PESO atual

MOV [R0], R1 ; Mover para a meta o valor do PESO atual

RET

; Escolha 2 - Hidratos

ChoiceDailyGoalChangeHidratos :

CMP R1, 2

JNE ChoiceDailyGoalChangeGordura ; Se não foi esta a
escolha do utilizador, verifica a próxima

; Input

MOV R1, 0

MOV R0, SEL_NR_MENU

MOV [R0], R1 ; Reset do periférico
[SEL_NR_MENU] antes de entrar no próximo menu

MOV R0, B_OK

MOV [R0], R1 ; Reset do periférico [B_OK]
antes de entrar no próximo menu

; Mudar valor da meta

MOV R0, META_HIDRATOS ; Mover para R0 o
valor da meta

MOV R1, PESO

MOV R1, [R1] ; Mover para R1 o valor do
PESO atual

MOV [R0], R1 ; Mover para a meta o valor
do PESO atual

RET

; Escolha 3 - Gordura

ChoiceDailyGoalChangeGordura :

CMP R1, 3

JNE ChoiceDailyGoalChangeCalorias ; Se não foi esta a
escolha do utilizador, verifica a próxima

; Input

MOV R1, 0

MOV R0, SEL_NR_MENU

MOV [R0], R1 ; Reset do periférico
[SEL_NR_MENU] antes de entrar no próximo menu

MOV R0, B_OK

MOV [R0], R1 ; Reset do periférico [B_OK]
antes de entrar no próximo menu

; Mudar valor da meta

MOV R0, META_GORDURA ; Mover para R0 o
valor da meta

MOV R1, PESO

MOV R1, [R1] ; Mover para R1 o valor do
PESO atual

MOV [R0], R1 ; Mover para a meta o valor
do PESO atual

RET

; Escolha 4 - Calorias

ChoiceDailyGoalChangeCalorias :

CMP R1, 4

JNE ChoiceDailyGoalChangeChoiceError ; Se não foi esta a
escolha do utilizador, verifica a próxima

; Input

MOV R1, 0

MOV R0, SEL_NR_MENU

MOV [R0], R1 ; Reset do periférico
[SEL_NR_MENU] antes de entrar no próximo menu

MOV R0, B_OK

MOV [R0], R1 ; Reset do periférico [B_OK]
antes de entrar no próximo menu

; Mudar valor da meta

MOV R0, META_CALORIAS ; Mover para R0 o
valor da meta

MOV R1, PESO

MOV R1, [R1] ; Mover para R1 o valor do
PESO atual

MOV [R0], R1 ; Mover para a meta o valor
do PESO atual

RET

; Erro

ChoiceDailyGoalChangeChoiceError :

; Input

MOV R3, 5

MOV R0, SEL_NR_MENU

MOV R1, B_OK

```
MOV                [R0], R3                ; Reset do periférico
[SEL_NR_MENU] antes de entrar no próximo menu

MOV                [R1], R3                ; Reset do periférico [B_OK]
antes de entrar no próximo menu

CALL               MenuChoiceErrorCall

RET
```

MenuDailyGoalCall:

```
; Display

CALL               PrepareDisplayCall        ; Após o botão de ligar ser
pressionado, preparar display para mostrar o menu principal

MOV                R2, GUIMenuDailyGoal      ; Guardar em R2 o
endereço do menu principal

CALL               DisplayMenuCall
```

MenuDailyGoalDisplayReady :

```
; Verificar se o utilizador confirmou a sua escolha

MOV                R0, B_OK

MOV                R1, [R0]                ; Escrever o valor de B_OK em
R1

CMP                R1, 1                    ; Verificar se o utilizador primiu
o botão de confirmar a escolha

JNE                MenuDailyGoalDisplayReady ; Se o utilizador não
cliquou confirmar a escolha ainda, voltar a verificar

; Se o utilizador confirmou a escolha

MOV                R0, SEL_NR_MENU

MOV                R1, [R0]                ; Escrever o valor de
SEL_NR_MENU em R1
```

; Escolha 1 - Alterar meta diária

CMP R1, 1

JNE ChoiceDailyGoalSee ; Se não foi esta a
escolha do utilizador, verifica a próxima

; Input

MOV R1, 0

MOV R0, SEL_NR_MENU

MOV [R0], R1 ; Reset do periférico
[SEL_NR_MENU] antes de entrar no próximo menu

MOV R0, B_OK

MOV [R0], R1 ; Reset do periférico [B_OK]
antes de entrar no próximo menu

CALL MenuDailyGoalChangeCall

RET

; Escolha 2 - Visualizar meta diária

ChoiceDailyGoalSee :

CMP R1, 2

JNE MenuDailyGoalError ; Se não foi esta a
escolha do utilizador, não existe mais nenhuma

; Input

MOV R1, 0

MOV R0, SEL_NR_MENU

```
MOV          [R0], R1          ; Reset do periférico
[SEL_NR_MENU] antes de entrar no próximo menu

MOV          R0, B_OK

MOV          [R0], R1          ; Reset do periférico [B_OK]
antes de entrar no próximo menu
```

```
CALL          MenuDailyGoalSeeCall
RET
```

; Escolha - Erro

```
MenuDailyGoalError      :
```

; Input

```
MOV          R3, 0
MOV          R0, SEL_NR_MENU
MOV          R1, B_OK
MOV          [R0], R3          ; Reset do periférico
[SEL_NR_MENU] antes de entrar no próximo menu

MOV          [R1], R3          ; Reset do periférico [B_OK]
antes de entrar no próximo menu

CALL          MenuChoiceErrorCall
RET
```

MenuResetCall:

; Display

```
CALL          PrepareDisplayCall          ; Preparar ecrã para
mostrar o menu balança
```



```
MOV          R2, GUIMenuReset          ; Guardar em R2 o
endereço do menu balança
CALL         DisplayMenuCall           ; Mostrar Menu balança
no Display
```

```
MenuResetDisplayReady          :
```

```
; Verificar se o B_OK foi pressionado
```

```
MOV          R0, B_OK
MOV          R0, [R0]
CMP          R0, 1
JNE          MenuResetOkNotPressed
```

```
; Se o B_OK foi pressionado
```

```
MOV          R0, SEL_NR_MENU
MOV          R0, [R0]
```

```
MenuResetChoiceReset          :
```

```
CMP          R0, 1
JNE          MenuResetChoiceReturnToMainMenu
```

```
MOV          R0, 0
MOV          R1, PROTEINA
MOV          [R1], R0          ; Reset proteina
MOV          R1, HIDRATOS
MOV          [R1], R0          ; Reset hidratos
MOV          R1, GORDURA
MOV          [R1], R0          ; Reset gordura
```

```
MOV            R1, CALORIAS
MOV            [R1], R0                ; Reset calorias

MOV            R0, 0
MOV            R1, B_OK
MOV            [R1], R0                ; Reset do B_OK
MOV            R1, SEL_NR_MENU
MOV            [R1], R0                ; Reset do SEL_NR_MENU

RET
```

MenuResetChoiceReturnToMainMenu :

```
CMP            R0, 2
JNE            MenuResetChoiceError

MOV            R0, 0
MOV            R1, B_OK
MOV            [R1], R0                ; Reset do B_OK
MOV            R1, SEL_NR_MENU
MOV            [R1], R0                ; Reset do SEL_NR_MENU

RET
```

MenuResetChoiceError :

```
MOV            R0, 0
MOV            R1, B_OK
MOV            [R1], R0                ; Reset do B_OK
MOV            R1, SEL_NR_MENU
MOV            [R1], R0                ; Reset do SEL_NR_MENU
```

Call MenuChoiceErrorCall

RET

; Se o B_OK não foi pressionado

MenuResetOkNotPressed :

JMP MenuResetDisplayReady

MenuChoiceErrorCall:

; Display

CALL PrepareDisplayCall ; Preparar Display para
mostrar o menu erro

MOV R2, GUIMenuChoiceError ; Guardar em R2 o
endereço do menu principal

CALL DisplayMenuCall

MenuChoiceErrorDisplayReady :

; Verifica se o botão B_OK foi pressionado

MOV R0, B_OK

MOV R0, [R0]

CMP R0, 1

JNE MenuChoiceErrorOkNotPressed

; Se o botão B_OK foi pressionado

MOV R0, 0

MOV R1, B_OK

MOV [R1], R0 ; Reset do B_OK

RET

; Se o botão B_OK não foi pressionado

MenuChoiceErrorOkNotPressed :

JMP MenuChoiceErrorDisplayReady

RET

MenuMainCall:

; Display

CALL PrepareDisplayCall ; Após o botão de ligar ser
pressionado, preparar display para mostrar o menu principal

MOV R2, GUIMenuMain ; Guardar em R2 o
endereço do menu principal

CALL DisplayMenuCall

MenuMainDisplayReady :

; Verificar se o utilizador desligou o botão On/Off

MOV R0, B_ON_OFF

MOV R0, [R0]

CMP R0, 1

JEQ ButtonOnOffTurnedOn

; Se o utilizador desligou o botão On/Off

RET

; Se o utilizador não desligou o botão On/Off

ButtonOnOffTurnedOn :

```
MOV                                R0, B_OK

MOV                                R1, [R0]                                ; Escrever o valor de B_OK em
R1

; Verificar se o utilizador confirmou a sua escolha

CMP                                R1, 1                                ; Verificar se o utilizador primiu
o botão de confirmar a escolha

JNE                                MenuMainDisplayReady                ; Se o utilizador não
cliqueu confirmar a escolha ainda, voltar a verificar

; Se o utilizador confirmou a escolha

MOV                                R0, SEL_NR_MENU

MOV                                R1, [R0]                                ; Escrever o valor de
SEL_NR_MENU em R1

; Escolha 1 - Menu Balança

CMP                                R1, 1

JNE                                ChoiceDailyTotal                    ; Se não foi esta a escolha
do utilizador, verifica a próxima

; Input

MOV                                R1, 0

MOV                                R0, SEL_NR_MENU

MOV                                [R0], R1                            ; Reset do periférico
[SEL_NR_MENU] antes de entrar no próximo menu

MOV                                R0, B_OK

MOV                                [R0], R1                            ; Reset do periférico [B_OK]
antes de entrar no próximo menu

MOV                                R0, PESO
```

```
MOV          R6, [R0]          ; Passar como parâmetro
valor do peso atual

MOV          R7, 0              ; Passar como parâmetro o
alimento 0

CALL         MenuScaleCall

JMP          MenuMainCall

; Escolha 2 - Total diário
ChoiceDailyTotal      :

CMP          R1, 2

JNE          ChoiceDailyGoal    ; Se não foi esta a escolha
do utilizador, verifica a próxima

; Input

MOV          R1, 0

MOV          R0, SEL_NR_MENU

MOV          [R0], R1          ; Reset do periférico
[SEL_NR_MENU] antes de entrar no próximo menu

MOV          R0, B_OK

MOV          [R0], R1          ; Reset do periférico [B_OK]
antes de entrar no próximo menu

CALL         MenuDailyTotalCall

JMP          MenuMainCall
```

; Escolha 3 - Meta diária

ChoiceDailyGoal :

CMP R1, 3

JNE ChoiceResetInput ; Se não foi esta a escolha
do utilizador, verifica a próxima

; Input

MOV R1, 0

MOV R0, SEL_NR_MENU

MOV [R0], R1 ; Reset do periférico
[SEL_NR_MENU] antes de entrar no próximo menu

MOV R0, B_OK

MOV [R0], R1 ; Reset do periférico [B_OK]
antes de entrar no próximo menu

CALL MenuDailyGoalCall

JMP MenuMainCall

; Escolha 4 - Reset

ChoiceResetInput :

CMP R1, 4

JNE MenuMainChoiceError ; Se não foi esta a
escolha do utilizador, não existe próxima, logo mostra um erro

; Display

CALL PrepareDisplayCall ; Preparar Display para
mostrar o menu de reset

MOV R2, GUIMenuReset ; Guardar em R2 o
endereço do menu de reset

CALL DisplayMenuCall ; Mostrar menu de reset
no Display

; Input

MOV R3, 0

MOV R0, SEL_NR_MENU

MOV R1, B_OK

MOV [R0], R3 ; Reset do periférico
[SEL_NR_MENU] antes de entrar no próximo menu

MOV [R1], R3 ; Reset do periférico [B_OK]
antes de entrar no próximo menu

CALL MenuResetCall

JMP MenuMainCall

; Erro

MenuMainChoiceError :

; Input

MOV R3, 0

MOV R0, SEL_NR_MENU

MOV R1, B_OK

MOV [R0], R3 ; Reset do periférico
[SEL_NR_MENU] antes de entrar no próximo menu

MOV [R1], R3 ; Reset do periférico [B_OK]
antes de entrar no próximo menu

CALL MenuChoiceErrorCall

JMP MenuMainCall