

Tpa trabalho

1- Criar um programa que leia o ano de nascimento de um usuário e o ano atual. Calcule e apresente a sua idade junto com mensagem correspondente: iv. Idade < 10 anos: criança; v. Idade ≥ 10 e < 18: Adolescente; vi. Idade ≥ 18 e Idade ≤ 60 : Idoso.

1-ler ano atual(at)
3-ler ano de nascimento(an)
4-idade($id \leftarrow at - an$)
5-ser($id < 10$)
6-então (“criança”)
7-ternão ser($id < 18$)
8-então (“adolescente”)
9-ternão ser ($id < 60$)
10-então (“adulto”)
11-ternão (“idoso”)

Programa:idade

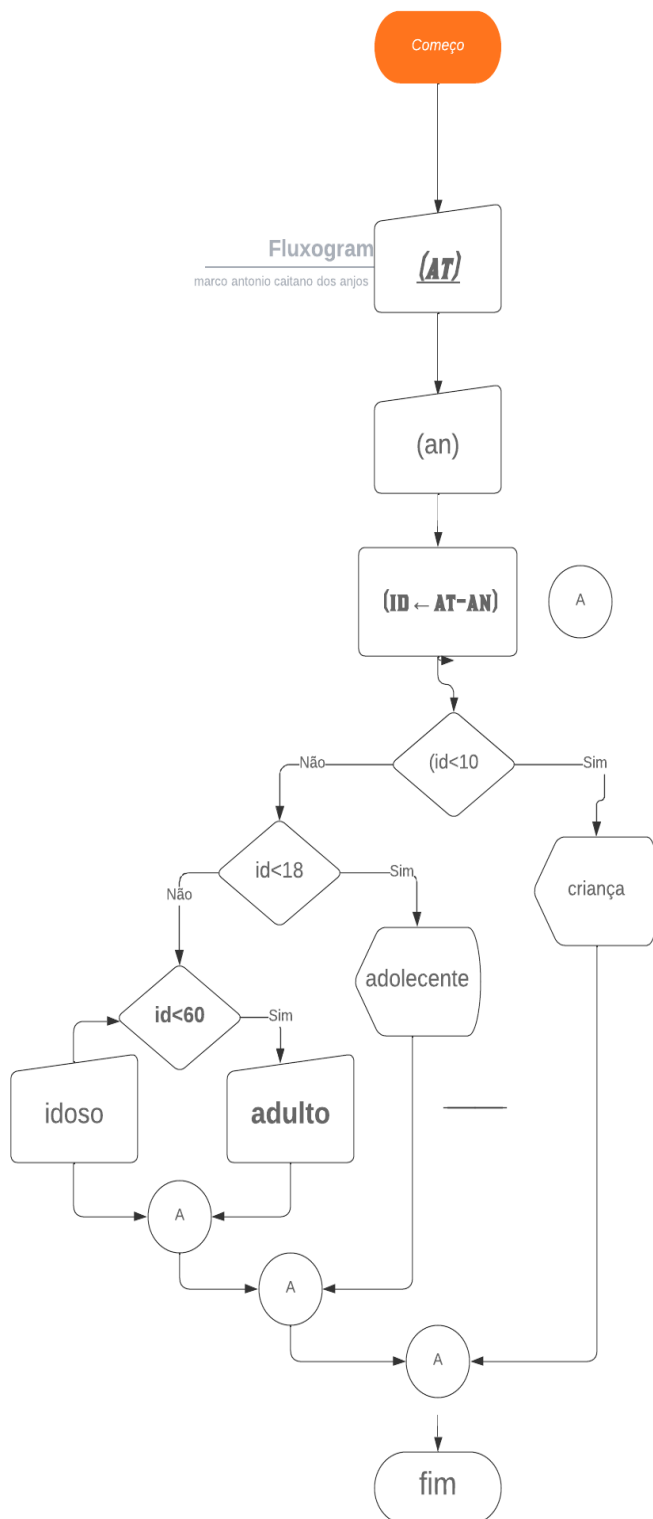
Var: at, na,

Início

leia(at)
leia(na)
 $id \leftarrow at - na$
ser($id < 10$) então
 escreva (“criança”)
ternão
 ser($id < 18$) então
 escreva (“adolescente”)

```
    sernão
    ser(id<60)
        escreva("adulto")
    sernão
        escreva("idoso")
fim_se
    fim_se
        fim_se
fim
```

Fluxogram
marco antonio caetano dos anjos.



Saiba mais sobre este modelo



2- Criar um programa que calcule e apresente o gasto médio de combustível (km por litro – km/L) de um veículo. O usuário deverá digitar a

distância percorrida (em Kilômetros) e a capacidade do tanque de combustível (em litros). O gasto médio é obtido dividindo-se a distância percorrida pela capacidade do tanque. Informar se o carro é econômico ou não, segundo a regra abaixo: a. Se o consumo for ≥ 10 km/L: Econômico b. Se o consumo < 10 km/L: Não econômico

- 1-leia distancia percorrida(dp)
- 2-leia capacidade do tanque(ct)
- 3-gasto médio($md \leftarrow dp/ct$)
- 4-ser($md \geq 10$) então
- 5-apresente (“econômico”)
- 6- Sernão
- 7-Ser($md < 10$) então
- 8- apresente(“não econômico”)

Programa: combustível

Var dp, ct, md

Inicia

Leia(dp)

Leia(ct)

$md \leftarrow ct/dp$

ser($md \geq 10$) então

 escreva(“econômico”)

 sernã

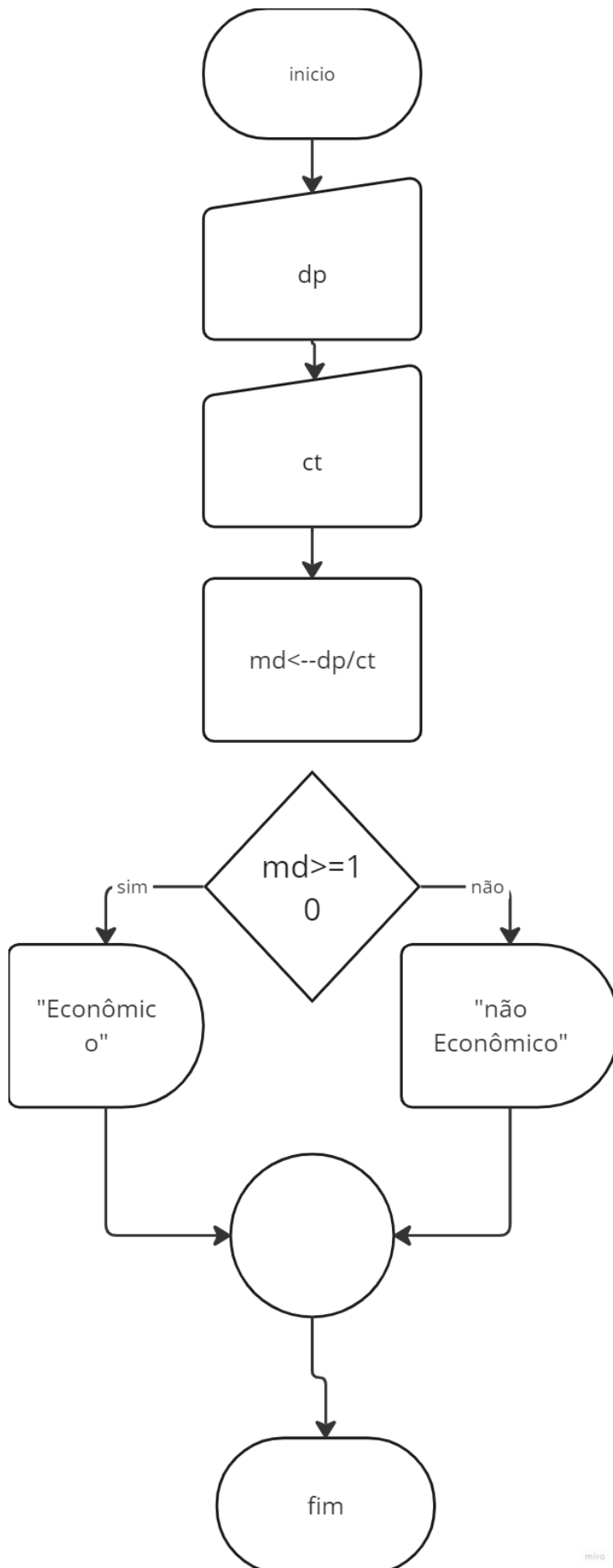
 ser($md < 10$) então

 escreva(“não econômico”)

fim_se

 fim _se

fin



Criar um programa que leia o peso e a altura de um usuário. Informar o seu IMC junto com a respectiva condição

1-leia peso(p)

2-leia altura(a)

3- seu imc ($\text{imc} \leftarrow p/a^2$)

4-ser ($\text{imc} < 18,5$) então

5-apresente (“excesso de magreza”)

6-ser não

7-ser($\text{imc} < 25$) então

Apresente(“peso normal”)

8-ser não

9-ser($\text{imc} < 30$) então

10-apresente(“excesso de peso”)

11-ser não

12-ser($\text{imc} < 35$) então

13-apresente(“obesidade grau 1”)

14-ser não

15- ser($\text{imc} < 40$) então

16-apresente("obesidade grau 2")

17-Sernão

18- Ser(imc>40) então

19-apresente("obesidade grau 3")

Programa:imc

Var p,a,imc

Inicio

Leia(p)

Leia(a)

Imc \leftarrow p/(a*a)

Ser imc<18,5 então

Escreva("excesso de magreza")

Sernão

Ser (imc<25) então

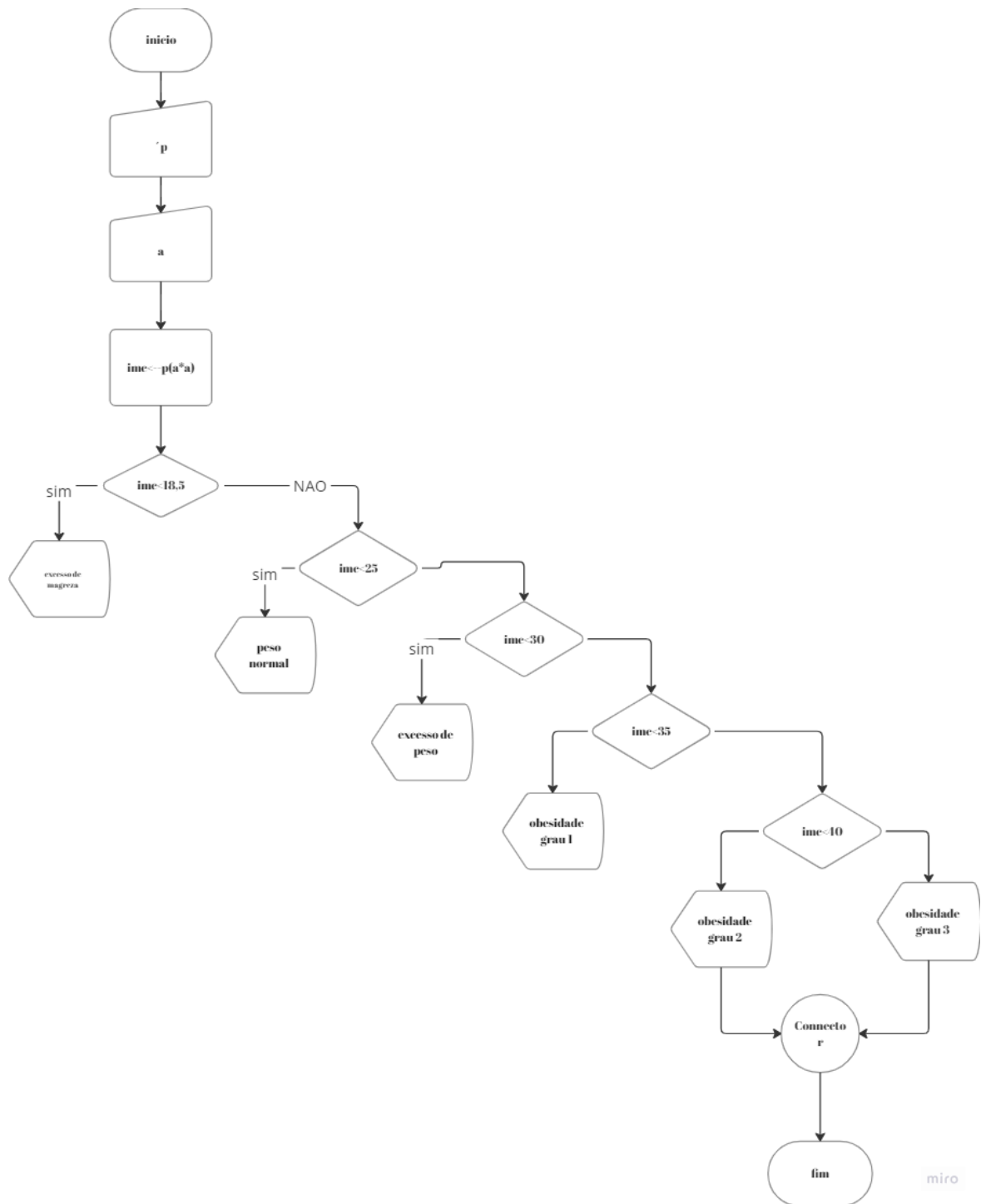
escreva("peso normal")

sernã

ser (imc<30) então

escreva("excesso de peso")

```
    sernão
  ser(imc<30) então
    escreva("excesso de peso")
  sernão
    ser(imc<35) então
      escreva("obesidade grau 1")
    sernão
      ser(imc<40) então
        escreva("obesidade grau
2")
      fim_se
    fim_se
  fim_ser
fim_se
fim_se
fim
```

4- Criar um programa que leia os valores A, B e C de uma equação de segundo grau (Ax^2+Bx+C). Calcular as duas raízes reais de X lembrando que: • Se $\Delta < 0$: não existem raízes reais para a equação; • Se $\Delta = 0$: existe apenas uma raiz real para a equação; • Se $\Delta > 0$: existem duas raízes reais para a equação.

1-leia a(a)

2-leia b(b)

3-leia c(c)

4-seu delta($d \leftarrow b*b - 4*a*c$)

5-ser($d < 0$) então

6-apresente("sem raízes")

7- sernão

8-ser($d = 0$) então

9-raiz $\leftarrow -b / 2*a$

10-apresenta("existe apenas uma raiz",
raiz)

11- sernão

12-raiz $\leftarrow -b + d / 2*a$

13-raiz2 $\leftarrow -b - d / 2*a$

14-apresente(“a duas raízes” raiz e raiz2)

Programa: raiz

Var

Int:

a,b,c,raiz,raiz2

Inicio

leia a(a)

leia b(b)

leia c(c)

seu delta($d \leftarrow b*b - 4*a*c$)

se($d < 0$) então

 escreva(“sem raízes”)

senão

 se($d = 0$) então

$raiz \leftarrow -b / 2*a$

 escreva(“existe apenas uma raiz”, raiz)

 senão

$raiz \leftarrow -b + d/2*a$

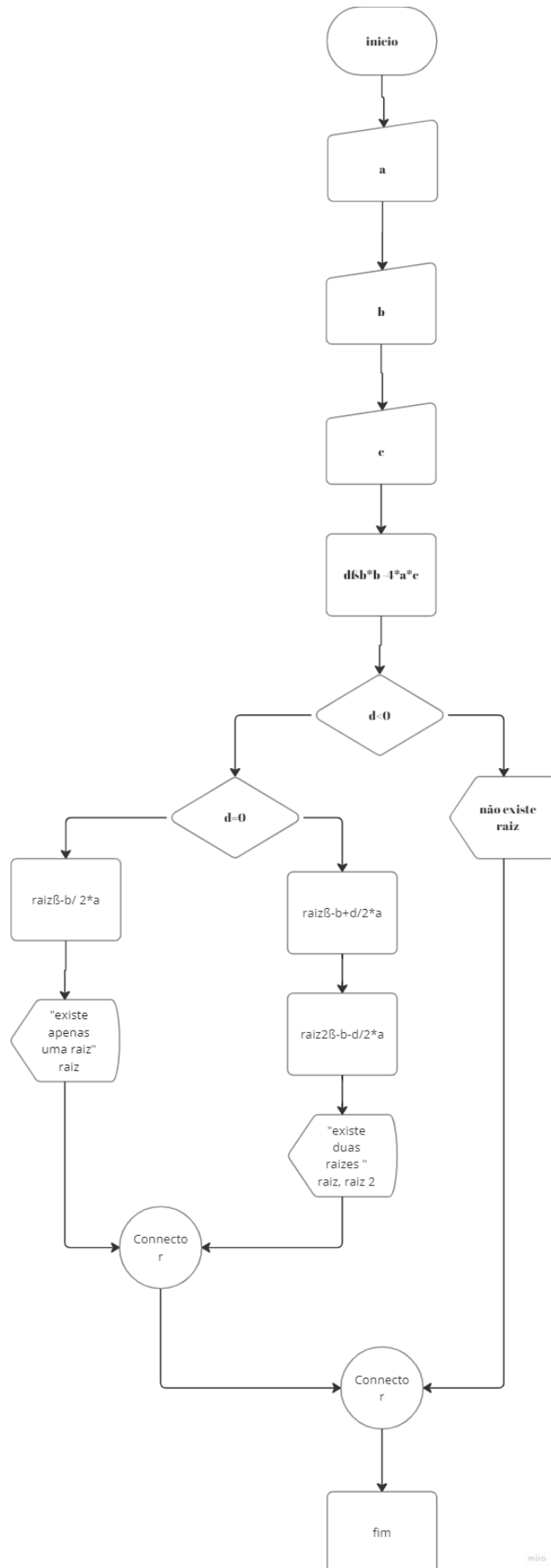
$raiz2 \leftarrow -b - d/2*a$

```
escreva("a duas raízes" raiz e raiz2)
```

```
    fim_se
```

```
    fim_se
```

```
fim
```



5- Criar um programa que leia 3 valores A, B e C para um possível triângulo e informe o tipo de triângulo lido. Verificar ainda se as medidas formam um triângulo segundo a lei: - Para que se possa formar um triângulo é necessário que a medida de qualquer um dos lados seja menor que a soma das medidas dos outros dois.

Triângulo Equilátero: Três lados iguais;

Triângulo Escaleno: Três lados diferentes;

Triângulos Isósceles: Dois lados iguais e um lado diferente.

1-ler numero A(a)

2-ler numero B(b)

3-ler numero C(c)

4- ser($a < b + c$, $b < a + c$, $c < a + b$)

5-ser ($a == b == c$) então

6-apresente('o triangulo é equilatero')

7-ternão

8-ser($a == b$ ou $b == c$ ou $a == c$) então

9-apresente("o triangulo é isosceleno")

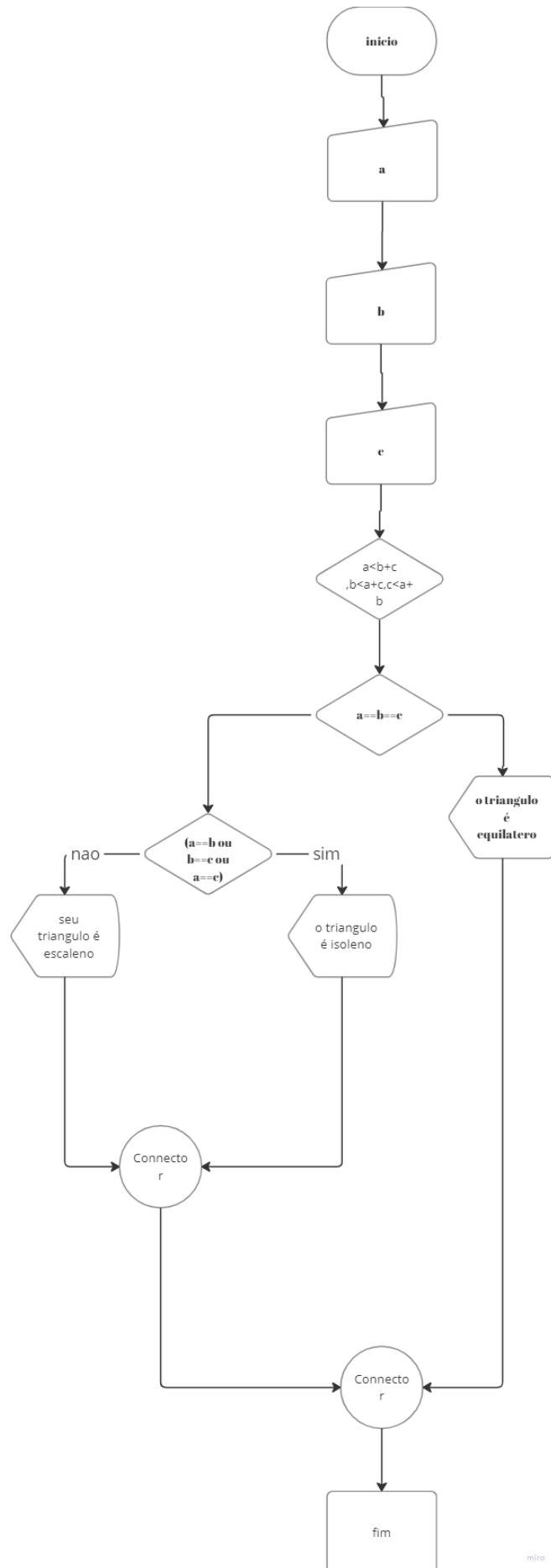
10-ternão

11-apresenta("seu triangulo é escaleno")

Programa:triangulo

```
Var  
int a,b,c  
inicio
```

```
ler numero A(a  
ler numero B(b)  
ler numero C(c)  
ser(  $a < b + c$  ,  $b < a + c$  ,  $c < a + b$  )  
ser (  $a == b == c$  ) então  
    escreva('o triangulo é equilatero')  
senão  
    ser( $a == b$  ou  $b == c$  ou  $a == c$ ) então  
        escreva("o triangulo é isosceles")  
    senão  
        escreva("seu triangulo é escaleno")  
    fim_se  
fim_se  
fim
```



6- Criar um programa que leia três valores inteiros A, B e C e os apresente em forma crescente (menor para o maior)

1-ler numero A(a)

2-ler numero B(b)

3-ler numero C(c)

4-se($a < b$) e ($a < c$)

5-(então) a=menor

6-se ($b < c$)

7-Então (b =meio) e (c =maior)

8-Senão

9- (c=meio) e (b=maior

10- Senão se (b<a) e (b<c)

11então (b=menor)

12-se (a<c)

13-então (a=meio) e (c=maior)

14-senão

15-(c=meio) e (a=maior)

16-senão se (c<a) e (c<b)

17-então (c=menor)

18-Se (a<b)

19-então (a=meio) e (b=maior)

20-senão

21-(a=maior) e (b=meio)

22-Apresenta(menor), (meio), (maior).

Programa:três_valores

Var:a,b,c,maior, menor, meio

ler numero A(a)

ler numero B(b)

ler numero C(c)

se($a < b$) e ($a < c$)

(então) $a = \text{menor}$

se ($b < c$)

Então ($b = \text{meio}$) e ($c = \text{maior}$)

Senão

($c = \text{meio}$) e ($b = \text{maior}$)

Senão se ($b < a$) e ($b < c$)

então ($b = \text{menor}$)

se ($a < c$)

($a = \text{meio}$) e ($c = \text{maior}$)

senão

15- ($c = \text{meio}$) e ($a = \text{maior}$)

senão se ($c < a$) e ($c < b$)

então ($c = \text{menor}$)

Se ($a < b$)

Então ($a = \text{meio}$) e ($b = \text{maior}$)

senão

($a = \text{maior}$) e ($b = \text{meio}$)

escreva(menor), (meio), (maior).

7- Criar um programa que leia o nome e a idade de 5 usuários. Ao final apresente o nome e a idade do usuário mais velho e o nome e a idade do usuário mais novo

1-Leia(nome)

2-leia(id)

3-Nomevelho \leftarrow nome

4-Idvelhor \leftarrow id

5-Nomenovo \leftarrow nome

7-leia (id)

8-leia(nome)

9-se (idvelho>id) então

11-nomevelhor \leftarrow nome

12-idvelhor \leftarrow id

13-ser não

14-ser(idnovo<id) então

15-idnovo \leftarrow id

16-nomenovo \leftarrow nome

17-Leia(id)

18-leia(nome)

19-ser(idvelhor>id) então

20- nomevelhor \leftarrow nome

21- idvelhor \leftarrow id

20-serve

21- se(idnovo<id) então

22- idnovo \leftarrow id

23- nomenovo \leftarrow nome

24-Leia(id)

25-leia(nome)

26-se(idvelhor>id) então

27- nomevelhor \leftarrow nome

28- idvelhor \leftarrow id

29-serve

30- se(idnovo<id) então

31- idnovo \leftarrow id

32- nomenovo \leftarrow nome

33-Leia(id)

34-leia(nome)

35-se(idvelhor>id) então

36- nomevelhor \leftarrow nome

37- idvelhor \leftarrow id

38-serve

39- se(idnovo<id) então

40- idnovo ← id

41- nomenovo ← nome

42-

apresenta(idnovo, nomenovo, nomevelho, idvelh
or

Programa: idade

Var

Int id, velhor, idnovo

Caractere: nome, nomevelho, nomenovo

Inicio

Leia(nome)

leia(id)

Nomevelho ← nome

Idvelhor ← id

Nomenovo ← nome

leia (id)

leia(nome)

(idvelho > id) então

nomevelhor ← nome

idvelhor ← id

se não

ser(idnovo<id) então

idnovo←id

nomenovo←nome

Leia(id)

leia(nome)

ser(idvelhor>id) então

nomevelhor←nome

idvelhor←id

ser não

ser(idnovo<id) então

idnovo←id

nomenovo←nome

Leia(id)

leia(nome)

ser(idvelhor>id) então

nomevelhor←nome

idvelhor←id

ser não

ser(idnovo<id) então

idnovo←id

nomenovo←nome

Leia(id)

leia(nome)

ser(idvelhor>id) então

nomevelhor←nome

idvelhor←id

ser não

ser(idnovo<id) então

idnovo←id

nomenovo←nome

escreva(idnovo,nomenovo,nomevelho,idvelhor

fim_se

fim_se

fim_se

fim_se

fim

6-- Criar um programa que leia duas notas para um aluno. Calcular e apresentar a sua

média, sendo que:

* SE a média for menor do que 3 o aluno está REPROVADO;

* SE a média for ≥ 6 o aluno está APROVADO;

* SE a média for ≥ 3 e < 6 avisar via mensagem que o aluno está em EXAME.

Solicitar

1-Leia nota1(n1)

2-leia nota 2(n2)

3-calcular media($m \leftarrow \frac{n1+n2}{2}$)

4-ser ($m < 3$) então

5- apresente("reprovado")

6-ser não

7-ser($m \geq 3$ e < 6) então

8-Apresente("esta em exame")

9-ser não

10-ser($m \geq 6$)

11-apresente("aprovado")

Programa:media

Var

Int:n1,n2,m

Inicio

Leia nota1(n1)

leia nota 2(n_2)

calcular media($m \leftarrow \frac{n_1 + n_2}{2}$)

se ($m < 3$) então

 apresente("reprovado")

senão

 se($m \geq 3$ e < 6) então

 Apresente("esta em exame")

 senão

 se($m \geq 6$)

 escreva("aprovado")

 fim_se

fim_se

fim

