



LINGUAGEM DE PROGRAMAÇÃO I:
LISTA DE EXERCÍCIOS I

Aluna: Claudia Barreto de Oliveira

Matricula: 20200019331

Matéria: Linguagem de Programação I – IC592

Período: 2020.5

Professor: Nilton José Rizzo

Rio de Janeiro, 21/10/2020.

Lista de exercícios.

(01)Faça para todos os exercícios, o algoritmo em linguagem natural, o fluxograma e o algoritmo estruturado. NÃO QUERO em portugol ou qualquer outro tipo de pseudo linguagem ou linguagem.

1.Calcule a soma dos N primeiros números ímpares digitado pelo usuário. O primeiro valor será atribuído a N, os demais serão somados, caso sejam ímpares.

2.Use os três tipos de controle de loop (FAÇA, FAÇA ENQUANTO,ENQUANTO FAÇA).

3.Calcule o N-ésimo valor da sequência de Fibonacci.

4.Calcule o N-ésimo valor da sequência $x \leftarrow 2N + 5$.

5.Calcule o fatorial de N.

6.Dado um valor inteiro, calcule os valores posterior e antecessor.

7.Dado duas variáveis inteiras, troque os valores das variáveis utilizando apenas essas duas variáveis.

8.O usuário irá digitar vários caracteres, e o programa deverá verificar se estes caracteres estão dentro da faixa (a – z, A – Z), e deve colocar em letras maiúsculas apenas a primeira letra de cada palavra. O programa termina quando o usuário digitar o carácter ponto(.). O programa deve ignorar qualquer carácter diferente da faixa permitida.

9.Uma empresa vende os produtos mostrados na tabela abaixo.

Código	Item	UNITARIO	>100	>500
001	Parafuso 1/8	0,10	8,00	6,50
002	Porca 1/8	0,05	4,50	4,00
003	Prego	0,10	9,00	8,00

Os valores unitário, e atacados com desconto. Faça um programa pra solicitar a lista de pedido de um cliente, e ao final mostre a lista com os valores respectivamente da quantidade comprada, valor a ser pago, seguindo a tabela e o total a ser pago.

10.Construa uma calculadora que execute as quatro operações básicas em sequência, e mostre o valor final quando for digitado a operação (=).

11.Construa um programa que aceite uma sequência infinita de valores inteiros e mostre qual o maior valor digitado a cada entrada. O programa termina quando o usuário digitar o valor 0 (zero)

1. Calcule a soma dos N primeiros números ímpares digitado pelo usuário. O primeiro valor será atribuído a N, os demais serão somados, caso sejam ímpares.

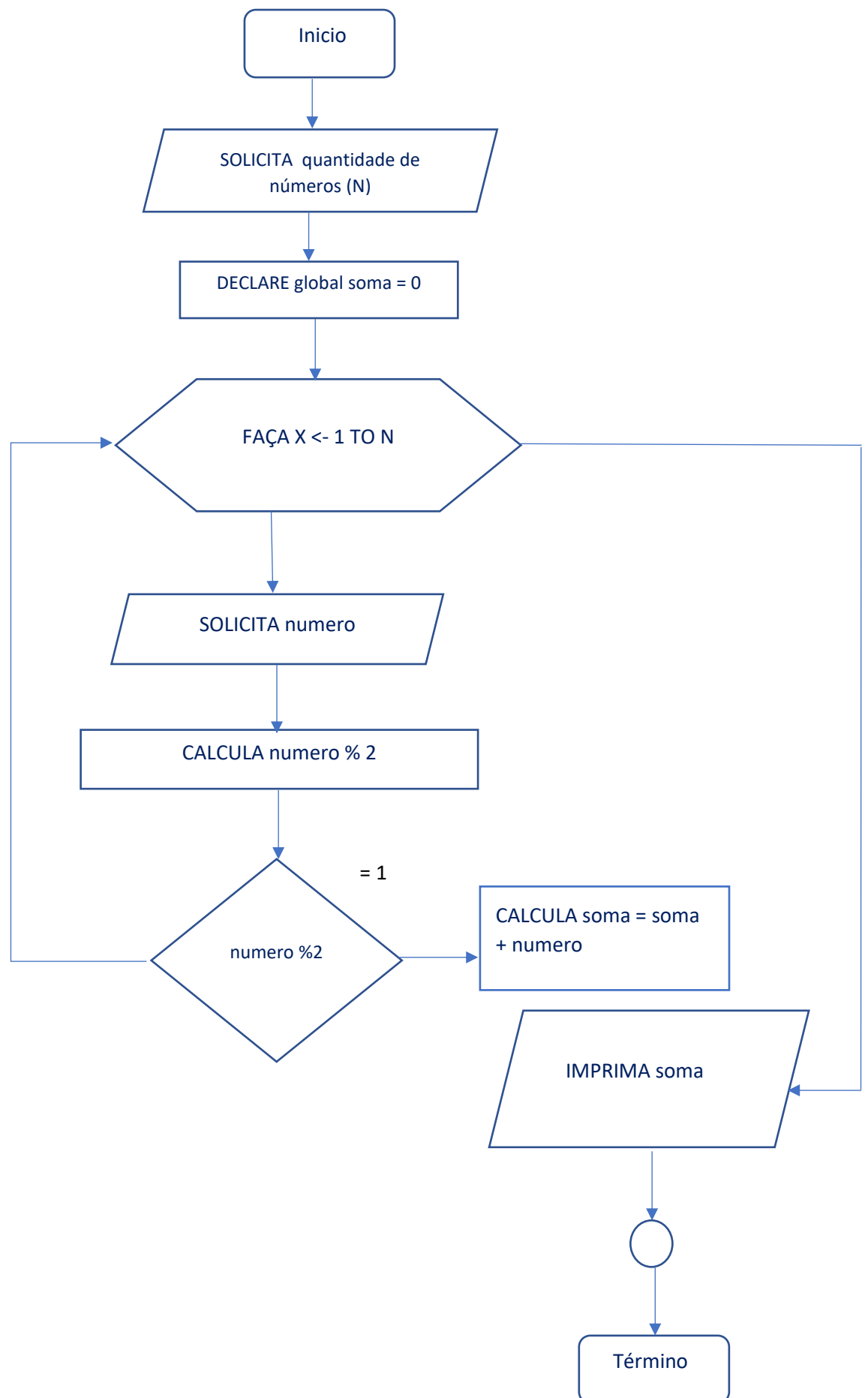
- LINGUAGEM NATURAL

- 1) INÍCIO - Soma dos N primeiros números ímpares
- 2) Solicitar a quantidade de números a serem imputados
- 3) Solicitar números enquanto $N \leq$ quantidade de números
 - 3.1) Calcular : $\text{numero} \% 2$
 - 3.2) SE valor calculado for igual a 1
 - 3.2.1) Efetuar a soma dos valores
- 4) Imprimir soma
- 5) Termina

- ALGORITMO ESTRUTURADO

- 1 – INICIO
- 2 – DECLARE: quantidade de números (N)
- 3 – INTEIRO SOMA
- 4 – ATRIBUA SOMA \leftarrow 0
- 5 – LEIA N
- 6 – FAÇA x \leftarrow 1 TO N:
 1. LEIA numero
 2. SE $\text{numero} \% 2 = 1$
 - 2.1 FAÇA $\text{soma} = \text{soma} + \text{numero}$
 3. FIM SE
- 7 – FIM FAÇA
- 8 – IMPRIMA soma
- 9- FIM

- FLUXOGRAMA



2. Use os três tipos de controle de loop (FAÇA, FAÇA ENQUANTO, ENQUANTO FAÇA).

A. Calcule o N-ésimo valor da sequência de Fibonacci.

B. Calcule o N-ésimo valor da sequência $x \leftarrow 2N + 5$.

C. Calcule o fatorial de N.

A) Calcule o N-ésimo valor da sequência de Fibonacci.

- LINGUAGEM NATURAL

1) Início - Fibonacci

2) Inteiros a, b, n, const, s,

3) Solicitar valor inicial da contagem (n)

4) Atribuir $a \leftarrow 1$

5) Atribuir $b \leftarrow 1$

6) Atribuir $const \leftarrow 2$

7) ENQUANTO $const < n$ FAÇA:

7.1) $s \leftarrow a + b$

7.2) $a \leftarrow b$

7.3) $b \leftarrow s$

7.4) $const \leftarrow const + 1$

8) fim ENQUANTO

9) Imprima n-ésimo valor da sequência de Fibonacci(b).

10) termino.

- ALGORITMO ESTRUTURADO

1 – INÍCIO

2 – INTEIRO a, b, n, const, s

3 – RECEBA a <- 1

4 –RECEBA b <- 1

5 –RECEBA const <- 2

6 – DECLARE VALOR n

7 – LEIA N

8 – ENQUANTO const < n FAÇA:

1. s <- a + b

2. a <- b

3. b <- s

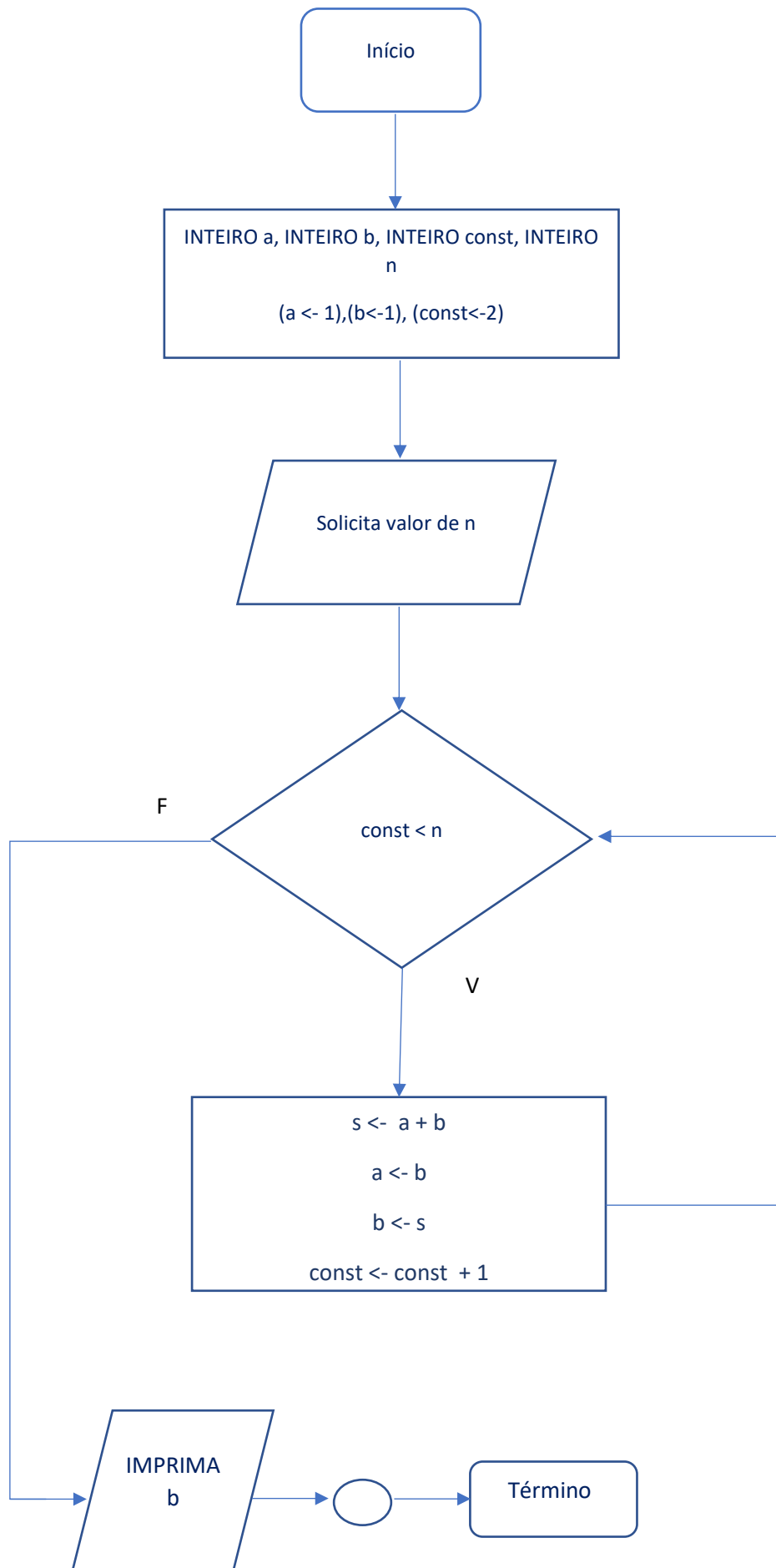
4. const <- const + 1

9 – FIM ENQUANTO

10 – MOSTRE b

11 – FIM.

- FLUXOGRAMA



B. Calcule o N-ésimo valor da sequência $x \leftarrow 2N + 5$.

- LINGUAGEM NATURAL

- 1) Inicio
- 2) $X = 2^N + 5$
- 3) Solicitar N, Limite
- 4) Atribuir Contador = 1
- 5) Fazer até contador = limite
 - 1) Ler N, contador
 - 2) Calcular $N \leftarrow N + 1$
 - 3) Calcular contador \leftarrow contador + 1
- 6) Calcule X
- 7) Imprimir X
- 8) Termine

- ALGORITMO ESTRUTURADO

- 1 – INICIO
- 2- DECLARE $X = 2^N + 5$
- 3 – DECLARE contador = 1
- 4 – DECLARE N, limite
- 5 – LEIA N, X, contador, limite
- 6 – FAÇA ATÉ que contador == limite
 - 6.1 LEIA N, contador
 - 6.2 CALCULE $N \leftarrow N + 1$
 - 6.3 CALCULE contador \leftarrow contador + 1
- 7 – FIM FAÇA
- 8 – CALCULE X
- 9 – LEIA X, N
- 10 – MOSTRE X
- 11 -FIM

C. Calcule o fatorial de N

- LINGUAGEM NATURAL

- 1) Inicio
- 2) Declarar o numero (n)
- 3) Escrever $n = n * (n-1) * (n-2) * (n-3)$
- 4) Calcular de F <- N ATÉ 1
- 5) Imprimir resultado
- 6) Termina

- ALGORITMO ESTRUTURADO

1 – INÍCIO

2 – DECLARE O NUMERO N

3 – LER N

4 – ESCREVER $N = N * (N-1) * (N-2) * (N-3)$

5 – FAÇA F <- N TO 1

5.1 - CALCULE ATÉ QUE $N = N * (N-1) * (N-2) * \dots * (N-(N-1))$

5.2 – MOSTRAR RESULTADO

6 – FIM FAÇA

7 – FIM

6. Dado um valor inteiro, calcule os valores posterior e antecessor.

- LINGUAGEM NATURAL

- 1) Início – Calcular valor sucessor e antecessor de um valor inteiro
- 2) Solicitar valor
- 3) Calcular sucessor \leftarrow valor + 1
- 4) Calcular antecessor \leftarrow valor – 1
- 5) Imprimir sucessor e antecessor
- 6) Término

- ALGORITMO ESTRUTURADO

1 – INÍCIO

2 – DECLARE valor

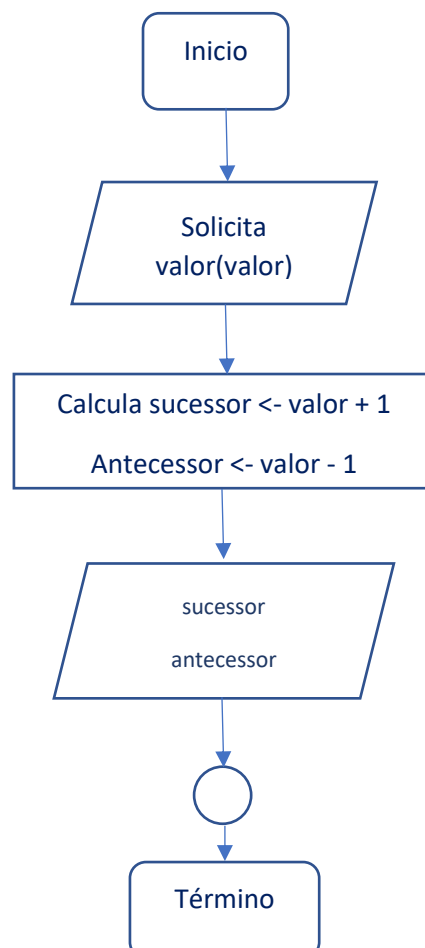
3 – CALCULE sucessor = valor + 1

4 – CALCULE antecessor = valor – 1

5 – MOSTRE O RESULTADO DE sucessor E antecessor

6 – FIM.

- FLUXOGRAMA



7. Dado duas variáveis inteiras, troque os valores das variáveis utilizando apenas essas duas variáveis.

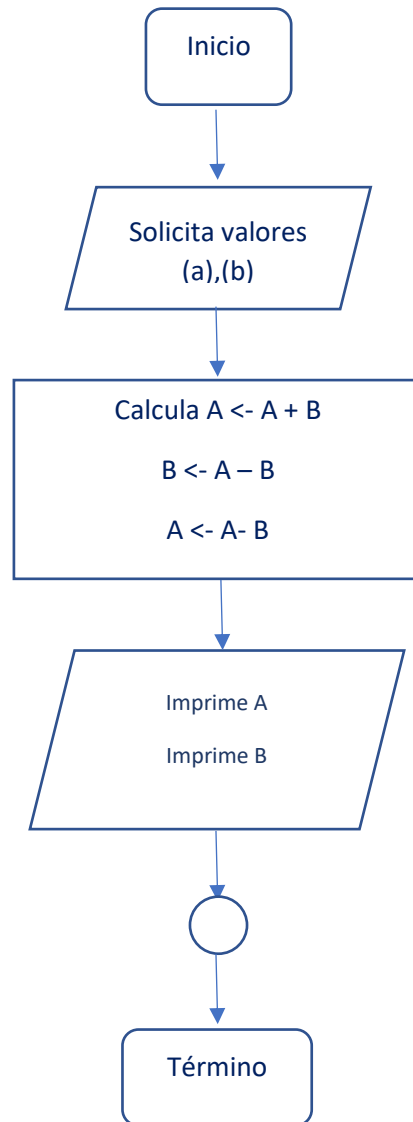
- LINGUAGEM NATURAL

- 1) Inicio – Trocar valor de duas variáveis
- 2) Solicitar valor de A
- 3) Solicitar valor de B
- 4) Calcular $A \leftarrow -A + B$
- 5) Calcular $B \leftarrow A - B$
- 6) Calcular $A \leftarrow A - B$
- 7) Imprima A
- 8) Imprima B
- 9) Termine.

- ALGORITMO ESTRUTURADO

- 1 – INICIO
- 2 – DECLARE A
- 3 – DECLARE B
- 4 – LEIA A, LEIA B
- 5 – CALCULE $A \leftarrow -A + B$
- 6 – CALCULE $B \leftarrow A - B$
- 7 – CALCULE $A \leftarrow A - B$
- 8 – MOSTRE A, MOSTRE B
- 9 – FIM

- FLUXOGRAMA



8.O usuário irá digitar vários caracteres, e o programa deverá verificar se estes caracteres estão dentro da faixa (a – z, A – Z), e deve colocar em letras maiúsculas apenas a primeira letra de cada palavra. O programa termina quando o usuário digitar o carácter ponto(.). O programa deve ignorar qualquer carácter diferente da faixa permitida

- LINGUAGEM NATURAL

- 1) Início
- 2) Solicitar uma entrada
 - 2.1) Se entrada <- faixa(a – z, A – Z)
 - 2.1.1) primeira letra maiúscula
 - 2.2) Usuário digitou “ ”(espaço)?
 - 2.2.1)Se sim, próxima letra será maiúscula
- 3) Repita o processo de solicitar até que seja digitado “.”
- 4) Imprima a entrada digitada
- 5) Termine

- ALGORITMO ESTRUTURADO

➔ Faixa ASCII

1 - INICIO

2 - DECLARE entrada

3 - ENQUANTO entrada != "."

1. LEIA entrada

2. SE entrada (>=65 e <=90) ou (>=97 e <=122) FAÇA

1. CASO entrada[0] (>=65 e <=90) FAÇA

1.1 entrada[0] <- entrada[0] - 32

2. FIM FAÇA

3. FIM SE

4. CASO entrada = 32 FAÇA

1. SE entrada[] (>=97 e <=122)

1.1 entrada[] <- entrada[] - 32

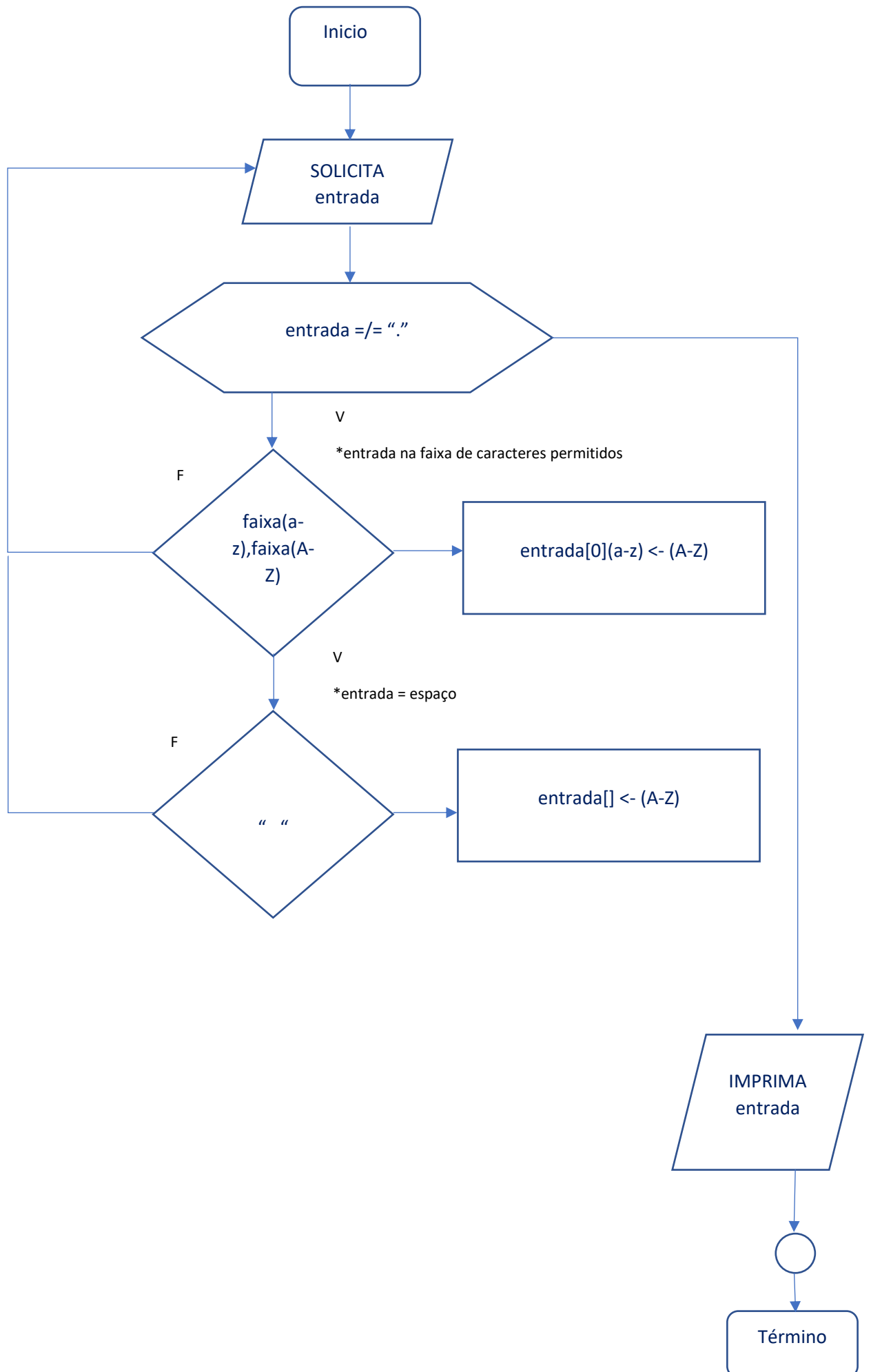
5. FIM FAÇA

4. FIM ENQUANTO

5. MOSTRE entrada

5. FIM

- FLUXOGRAMA



- RASCUNHO DA 8

- 1 – INICIO
- 2 - ENQUANTO SOLICITAR entrada :
 1. Se entrada[0] <- faixa(a – z, A – Z) ENTÃO FAÇA: *ou seja vai ser valida
 - 1.1 primeira letra maiúscula
- 3- CASO CONTRÁRIO Repita o processo de solicitar
 1. Usuário digitou “ ”(espaço)?
 - 1.1 Se sim, próxima letra será maiúscula
- 6) Repita o processo de solicitar até que seja digitado “.”
- 7) Fim

- 1 – INICIO
- 3- TEXTO faixa_um = (65,90)
- 4-TEXTO faixa_dois = (97,122)
- 2 – DECLARE entrada
- 5 – ENQUANTO entrada != “.”
 1. LEIA entrada
 2. SE entrada == faixa_um
 - 1.1 SE entrada[0] <- (faixa_um) FAÇA:
 - 1.2 entrada[0] <- (faixa_dois)
 3. FIM SE
 4. CASO entrada = “ ” FAÇA
 - 1.1 entrada[] <- (faixa_dois)
- 6- FIM ENQUANTO
- 7- FIM.

- 1 – INICIO
- 2 – DECLARE entrada
- 5 – ENQUANTO entrada != “.”
 5. LEIA entrada
 6. SE entrada (>= 65 e <= 90) ou (>= 97 e <= 122)
 - 1.3 SE entrada[0] (>= 65 e <= 90) FAÇA:

1.4 entrada[0] <- entrada[0] - 32

7. FIM SE

8. CASO entrada = 32 FAÇA

1.2 SE entrada[] (≥ 97 e ≤ 122)

1.2.1 FAÇA entrada[] <- entrada[] - 32

2. FIM SE

6- FIM ENQUANTO

7- FIM.

9. Uma empresa vende os produtos mostrados na tabela abaixo.

Código	Item	UNITARIO	>100	>500
001	Parafuso 1/8	0,10	8,00	6,50
002	Porca 1/8	0,05	4,50	4,00
003	Prego	0,10	9,00	8,00

Os valores unitário, e atacados com desconto. Faça um programa pra solicitar a lista de pedido de um cliente, e ao final mostre a lista com os valores respectivamente da quantidade comprada, valor a ser pago, seguindo a tabela e o total a ser pago.

- LINGUAGEM NATURAL

- 1) Inicio
- 2) Solicitar QUANTIDADE DE PARAFUSOS, QUANTIDADE DE PORCAS, QUANTIDADE DE PREGOS
- 3) Calcular total = $(\text{quantidade de parafusos} / 500) * 6.5 + (\text{quantidade de parafusos} - (\text{quantidade de parafusos} / 500) * 500) * 8 + \text{quantidade de parafusos} \% 100 * 0.1) + (\text{quantidade de porcas} / 500) * 4 + ((\text{quantidade de porcas} / 500) * 500) * 4.5 + \text{quantidade de porcas} \% 100 * 0.05 + ((\text{quantidade de pregos} / 500) * 8) + ((\text{quantidade de pregos} / 500) * 500) / 100 * 9 + \text{quantidade de pregos} \% 100 * 0.1$
- 4) Imprimir total
- 5) Termina

- ALGORITMO ESTRUTURADO

1 – INICIO

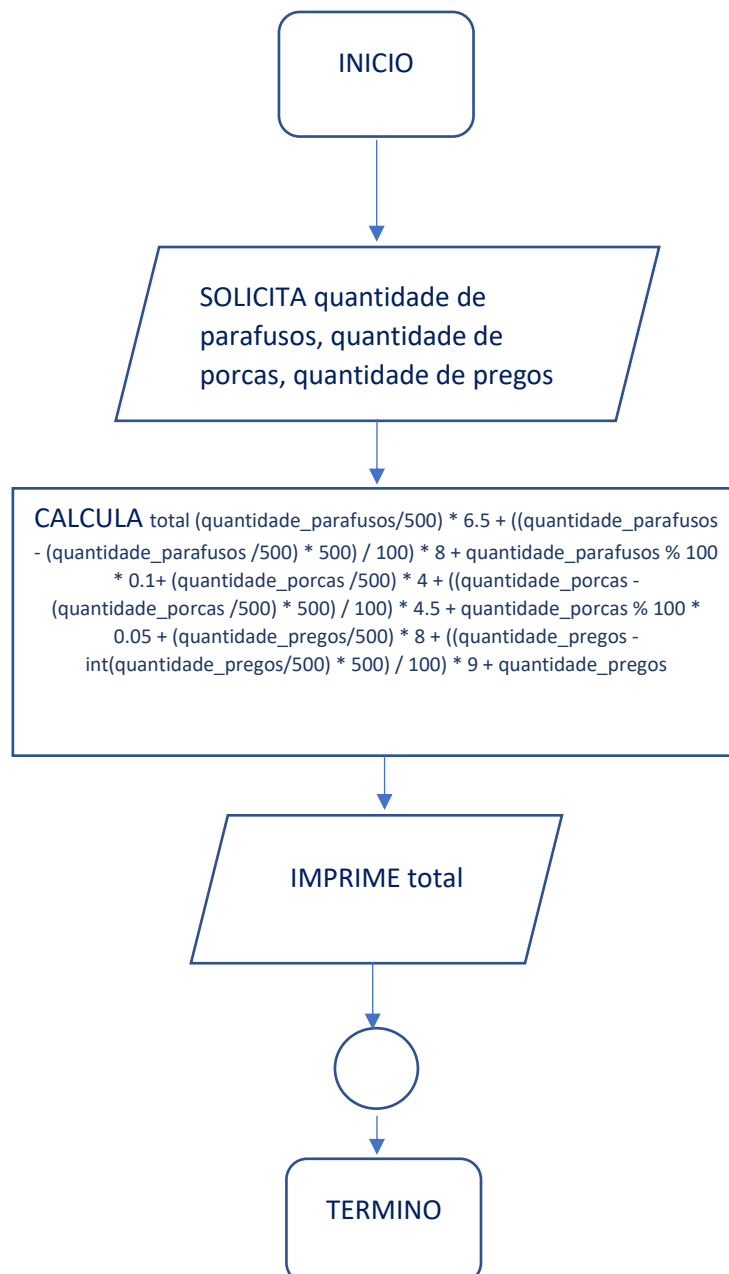
2 – DECLARE quantidade_parafusos, quantidade_porcas, quantidade_pregos

3 – CALCULE $total = (quantidade_parafusos / 500) * 6.5 + ((quantidade_parafusos - (quantidade_parafusos / 500) * 500) / 100) * 8 + quantidade_parafusos \% 100 * 0.1 + (quantidade_porcas / 500) * 4 + ((quantidade_porcas - (quantidade_porcas / 500) * 500) / 100) * 4.5 + quantidade_porcas \% 100 * 0.05 + (quantidade_pregos / 500) * 8 + ((quantidade_pregos - int(quantidade_pregos / 500) * 500) / 100) * 9 + quantidade_pregos \% 100 * 0.1$

4 - MOSTRE total

5 – FIM

- FLUXOGRAMA



10. Construa uma calculadora que execute as quatro operações básicas em sequência, e mostre o valor final quando for digitado a operação (=).

- LINGUAGEM NATURAL

- 1) Inicio
- 2) Solicitar numero
- 3) Declarar entrada
- 4) Declarar sinal (+, -, *, /)
- 5) Enquanto entrada \neq "="
 1. Se entrada = (+, -, *, /) : sinal = entrada
 2. Se entrada \neq (+, -, *, /):
 - 2.1 Se sinal = "+": vai calcular a soma
 - 2.2 Se sinal = "-": vai calcular subtração
 - 2.3 Se sinal = "*": vai calcular multiplicação
 - 2.4 Se sinal = "/": vai calcular divisão
- 6) Mostre resultado
- 7) Termina

- ALGORITMO ESTRUTURADO

1 – INICIO

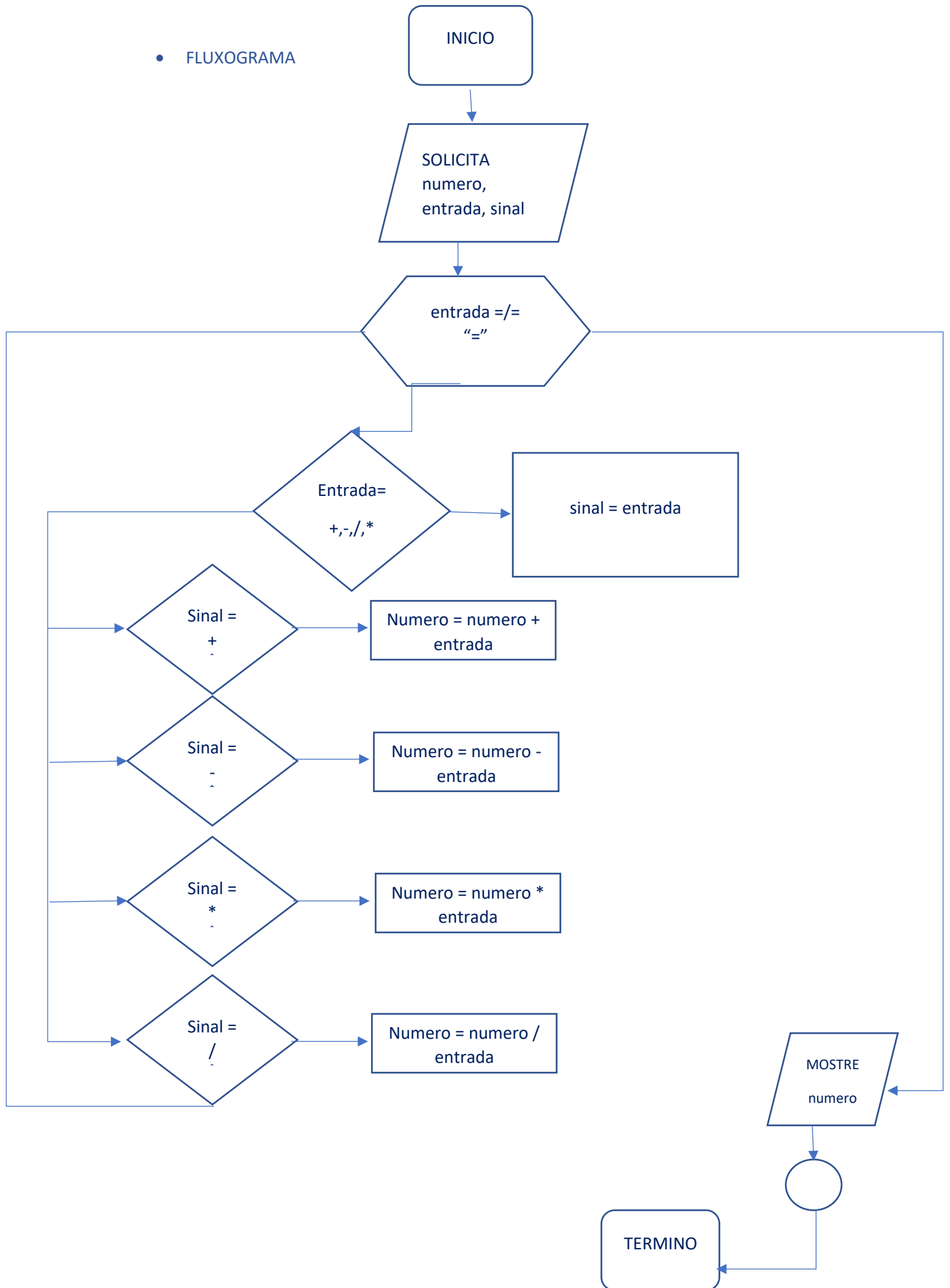
- 2 – INTEIRO entrada
- 3 – Declare numero
- 4 – Declare sinal
- 5 – ENQUANTO entrada \neq "=" FAÇA
 1. SE entrada = "+", "-", "*", "/"
 - 1.1 sinal = entrada
 2. SE NÃO FAÇA:
 - 2.1 sinal = "+"
 - 2.1.1 numero = numero + entrada
 - 2.2 sinal = "-"
 - 2.2.1 numero = numero – entrada
 - 2.3 sinal = "*"
 - 2.3.1 numero = numero * entrada
 - 2.4 sinal = "/"
 - 2.4.1 numero = numero / entrada
 3. FIM SE

4 – FIM ENQUANTO

5- MOSTRE numero

6- FIM

- FLUXOGRAMA



11. Construa um programa que aceite uma sequência infinita de valores inteiros e mostre qual o maior valor digitado a cada entrada. O programa termina quando o usuário digitar o valor 0 (zero)

- LINGUAGEM NATURAL

- 1) Inicio
- 2) Solicitar entrada
- 3) Declarar maior = 0
- 4) Enquanto entrada \neq 0 e entrada > maior
- 5) maior = entrada
- 6) Imprima maior
- 7) Fim

- ALGORITMO ESTRUTURADO

1 - Inicio

2- DECLARE entrada

3- INTEIRO maior = 0

4- ENQUANTO entrada \neq 0 FAÇA

1. SE entrada > maior FAÇA
 - 1.2 maior = entrada
2. FIM SE
3. MOSTRE maior

5- FIM ENQUANTO

6- TERMINO

- FLUXOGRAMA

