

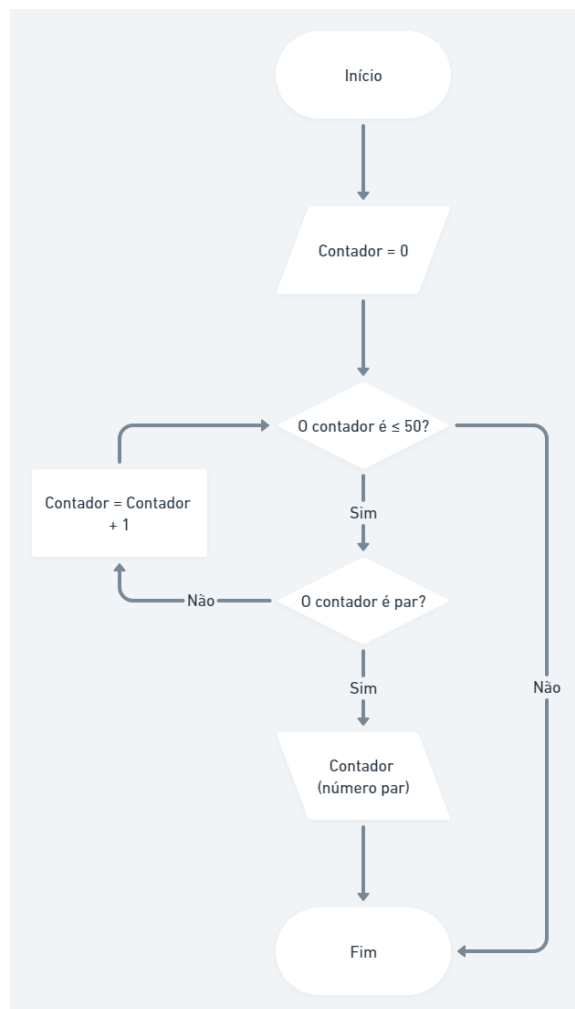


Trabalho 2º bimestre

Wesley Bernardes de Souza Júnior (00020321).

1) Criar um programa que mostre os valores pares no intervalo de 0 ate 50.

Fluxograma:



Código:

```
Var  
    contador: inteiro
```

```

Início
  contador <- 0

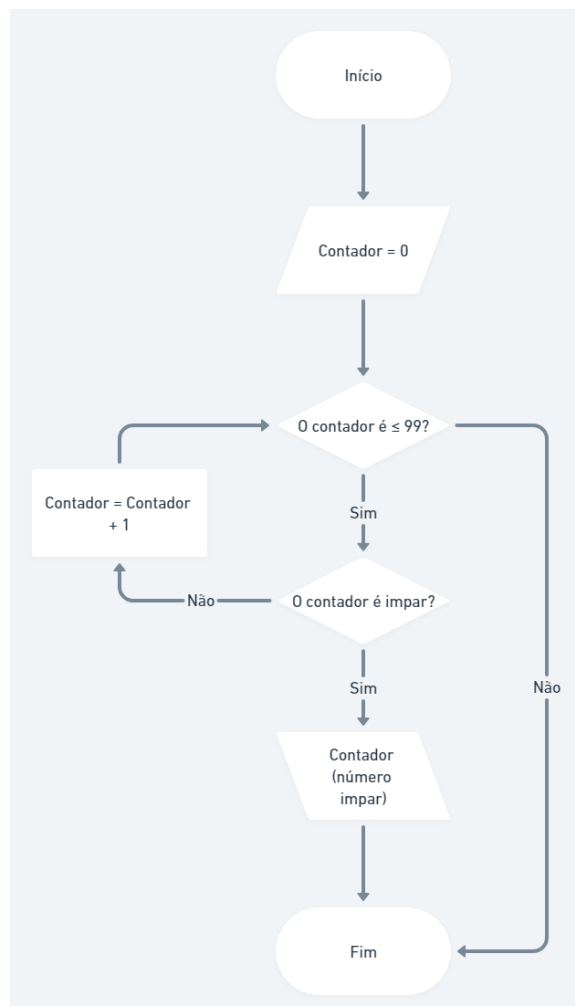
  enquanto (contador <=50) faça
    se (contador % 2 = 0) então
      escreval("O número é par: ", contador)
    fimse

    contador <- contador + 1
  fimenquanto
Fimalgoritmo

```

2) Criar um programa que mostre os valores impares no intervalo de 0 ate 99.

Fluxograma:



Código:

```
Var
  contador: inteiro

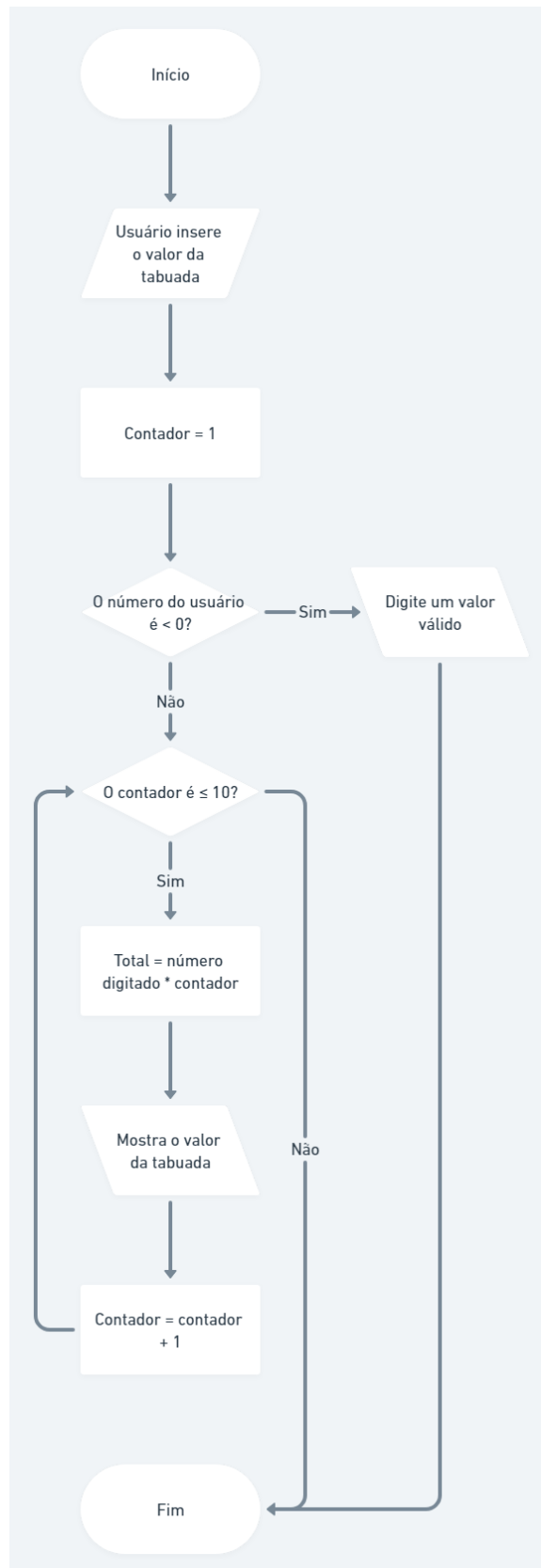
Inicio
  contador <- 0

  enquanto (contador <= 99) faca
    se (contador % 2 = 1) entao
      escreval(contador)
    fimse

    contador <- -contador+1
  fimenquanto
Fimalgoritmo
```

3) Criar um programa que o usuário entre com 1 valor e realizar e mostra a tabuada (Valor inteiro e positivo).

Fluxograma:



Código:

```
Var
    contador, numero, total: inteiro

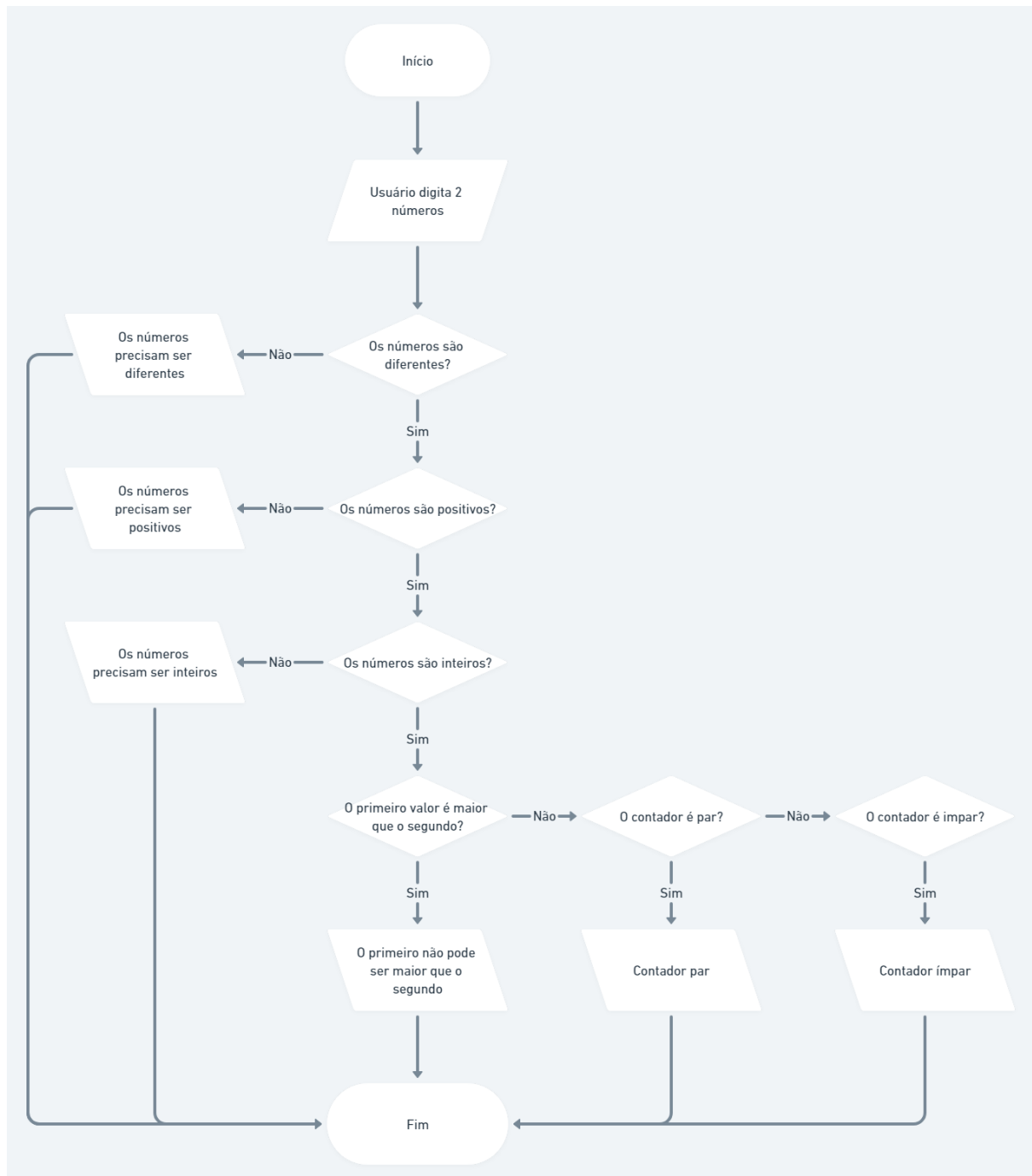
Inicio
    escreva("escreva um numero de 1 a 10: ")
    leia(numero)

    para contador de 0 ate 10 faca
        total <- numero * contador
        escreval(contador, " *", numero, " =", total)
    Fimpara

Fimalgoritmo
```

4) Criar um programa que o usuário entre com os valores início e fim do intervalo (1 valor não pode ser maior que o segundo e ambos tem que ser inteiros e positivos e diferentes), e mostrar quantos números pares e quantos números ímpares existem no intervalo.

Fluxograma:



Código:

```

Var
    primeiroNumero, segundoNumero, numerosImpares, numerosPares: inteiro

Início
    escreval("Digite o primeiro valor: ")
    leia(primeiroNumero)

    escreval("Digite o segundo valor: ")
    leia(segundoNumero)
  
```

```

se (primeiroNumero < segundoNumero) entao
  se ((primeiroNumero > 0) e (segundoNumero > 0)) entao
    enquanto (primeiroNumero < segundoNumero) faca
      primeiroNumero <- primeiroNumero + 1

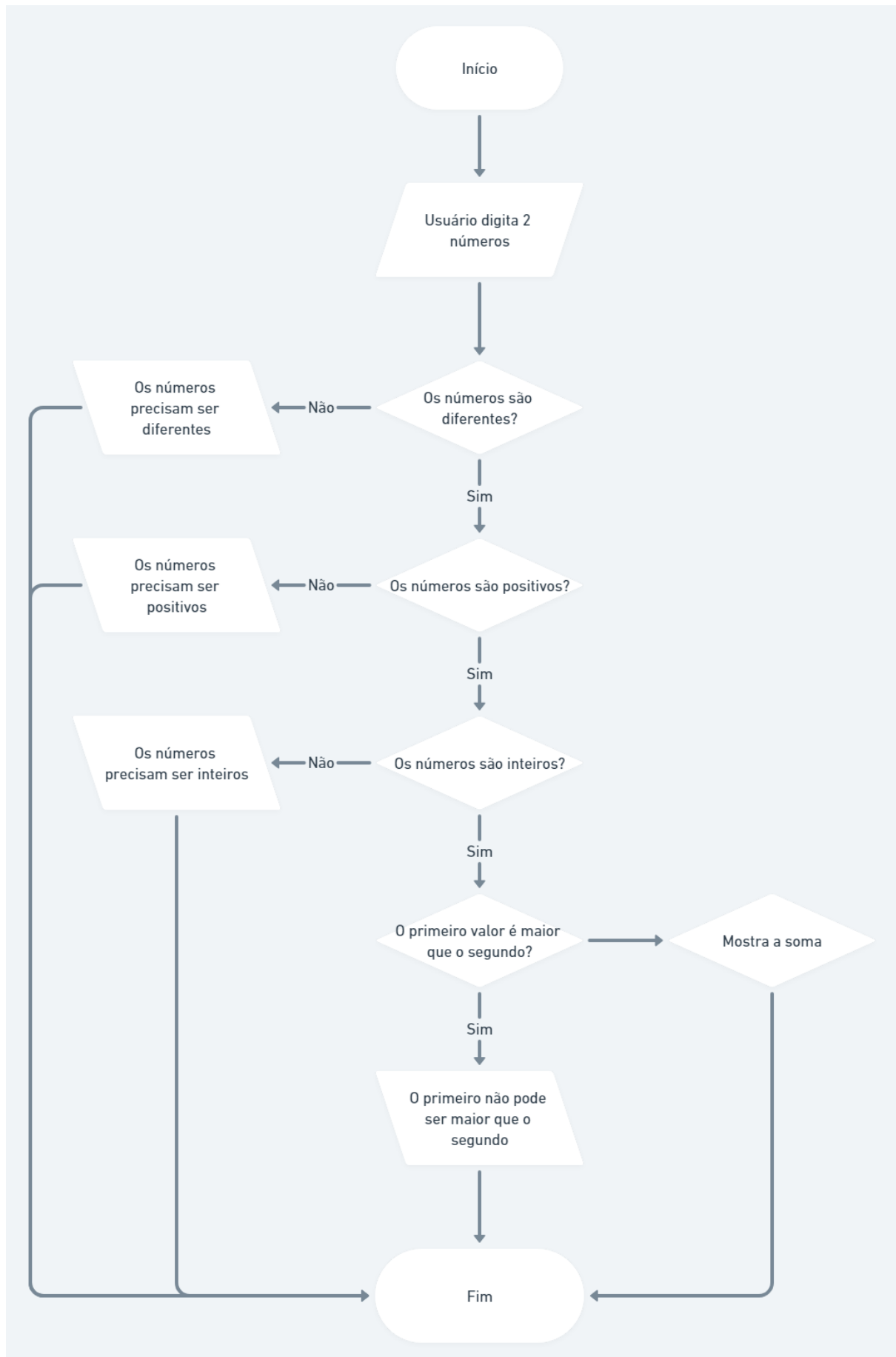
      se ((primeiroNumero % 2) = 1) entao
        numerosImpares <- numerosImpares + 1
      senao
        numerosPares <- numerosPares + 1
      fimse
    fimenquanto

    escreval("Os números pares são: ", numerosPares)
    escreval("Os números ímpares são: ", numerosImpares)
  senao
    escreval("Os números devem ser positivos.")
  fimse
senao
  escreval("O primeiro número não pode ser maior que o segundo.")
fimse
Fimalgoritmo

```

5) Criar um programa que o usuário entre com os valores início e fim do intervalo (1 valor não pode ser maior que o segundo e ambos tem que ser inteiros e positivos e diferentes), e realizar e mostrar o somatório do intervalo.

Fluxograma:



Código:

```
Var
    numero1, numero2, soma: inteiro

Inicio
    escreva("Insira o primeiro numero: ")
    leia(numero1)

    escreva("Insira o segundo numero: ")
    leia(numero2)

    soma <- 0

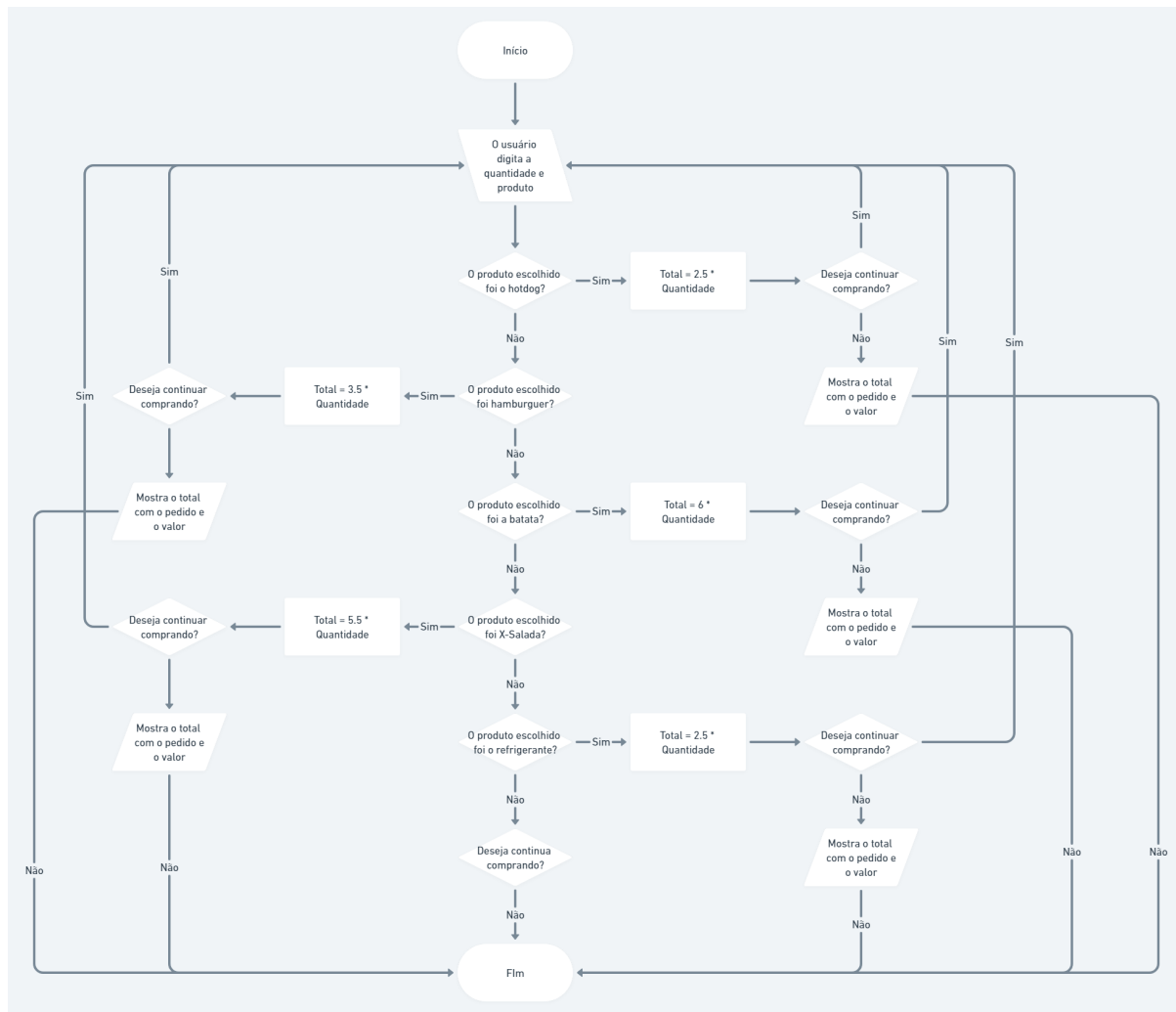
    se (numero1 < numero2) entao
        se ((numero1 > 0) e (numero2 > 0)) entao

            enquanto (numero1 <= numero2) faca
                soma <- soma + numero1
                numero1 <- numero1 + 1
            fimenquanto

            escreval("A soma deu: ", soma)
        senao
            escreva("Os números devem ser positivos.")
        fimse
    senao
        escreva("O primeiro número não pode ser maior que o segundo.")
    fimse
fimalgoritmo
```

6) Criar um programa que o usuário entre com a quantidade e qual produto ele deseja comprar quantas vezes quiser e assim que ele sair do programa mostrar o total do pedido do cliente.

Fluxograma:



Código:

```

Var
    numeroPedido, quantidadePorPedido: inteiro
    totalPedido: real

Inicio
    enquanto (numeroPedido <> 6) faca
        escreval("Tabela dos pedidos")
        escreval("1 - Hotdog R$ 2,5")
        escreval("2 - Hamburguer R$ 3,5")
        escreval("3 - Batata R$ 6,0")
        escreval("4 - X-salada R$ 5,5")
        escreval("5 - Refrigerante R$2,5")
        escreval("6 - Sair")
        leia(numeroPedido )

        se (numeroPedido <> 6) entao
            escreval("Digite a quantidade que você deseja")
            leia(quantidadePorPedido)
        fimse
    
```

```
escolha numeroPedido
  caso 1
    totalPedido <- totalPedido + quantidadePorPedido * 2.5

  caso 2
    totalPedido <- totalPedido + quantidadePorPedido * 3.5

  caso 3
    totalPedido <- totalPedido + quantidadePorPedido * 6

  caso 4
    totalPedido <- totalPedido + quantidadePorPedido * 5.5

  caso 5
    totalPedido <- totalPedido + quantidadePorPedido * 2.5

  caso 6
    escreval ("O valor total do pedido foi de: R$ ", totalPedido:8:2)
  fimescolha
fimenquanto
fimalgoritmo
```