

1) Sabendo-se que o polinômio: $ax^2 + bx + c = 0$ é resolvido com a fórmula de Bhaskara, peça ao usuário os coeficientes a, b e c e calcule as raízes do polinômio. O programa deve para caso o usuário solicite, após o cálculo das raízes, digitando a tecla <s>, tanto em maiúscula quanto em minúscula.

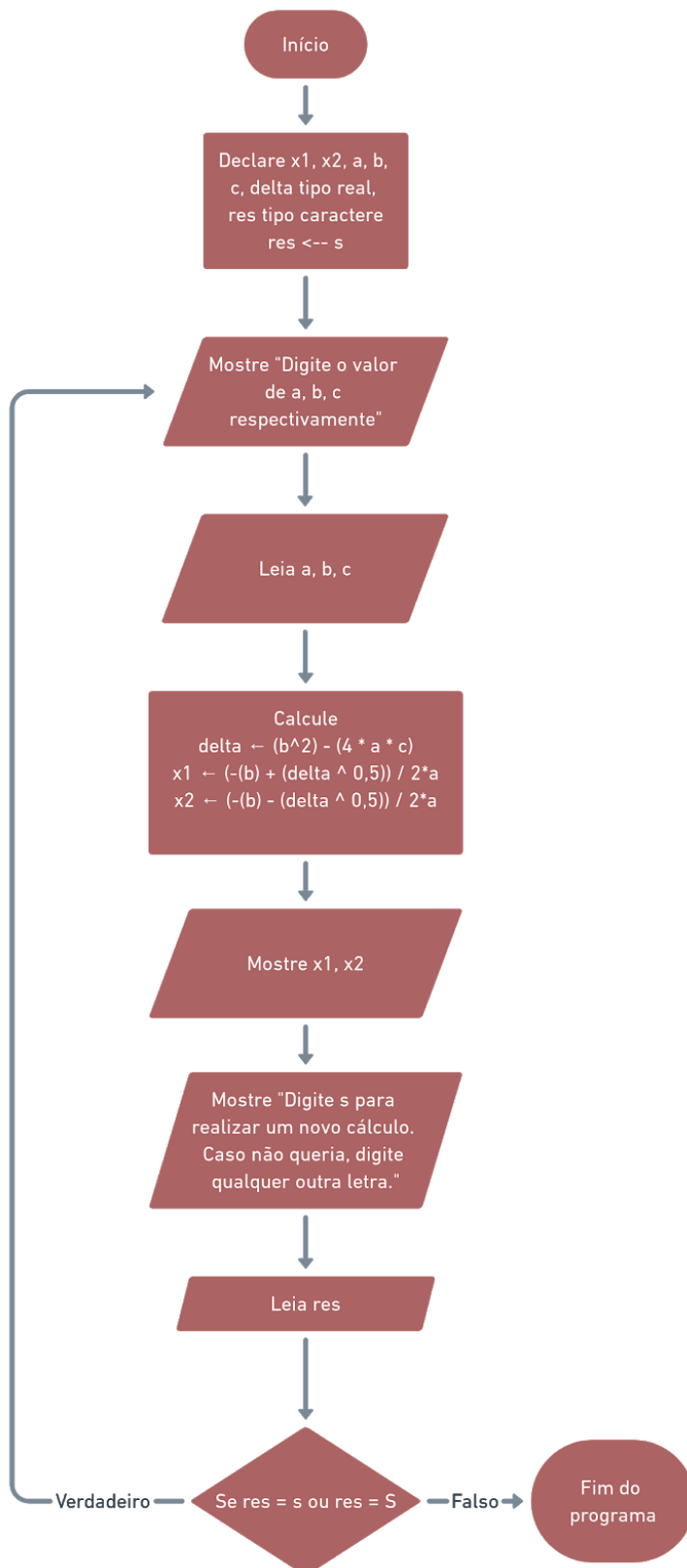
Linguagem Natural

1. Início
2. Solicite aos usuário o coeficiente a, b e c
3. Calcule $x = \frac{-b \pm (\sqrt{b^2 - 4 \cdot a \cdot c})}{2 \cdot a}$
4. Mostre o resultado ao usuário
5. Pergunte ao usuário se deseja calcular outro polinômio
6. Se sim, retorne ao passo 2. Se não, encerre o programa
7. Fim do programa

Linguagem Estruturada

1. Início
2. Declare x1, x2, a, b, c, delta tipo inteiro, res tipo caractere
3. $res \leftarrow res$
4. Enquanto $res \neq s$ OU $res \neq S$, faça
 - 4.1. Mostre "Digite o valor de a, b e c respectivamente:"
 - 4.2. Leia a
 - 4.3. Leia b
 - 4.4. Leia c
 - 4.5. Calcule
 - 4.5.1. $delta \leftarrow (b^2) - (4 * a * c)$
 - 4.5.2. $x1 \leftarrow \frac{-(b) + (delta^{0,5})}{2*a}$
 - 4.5.3. $x2 \leftarrow \frac{-(b) - (delta^{0,5})}{2*a}$
 - 4.6. Mostre x1, x2
 - 4.7. Mostre "Digite s para realizar um novo cálculo. Caso não queria, insira qualquer outra letra."
 - 4.8. Leia res
5. Fim enquanto
6. Fim do programa

Fluxograma



2) Elabore um algoritmo onde o usuário fornecerá a altura da árvore e o caractere para a sua construção, sabendo-se que: a altura mínima da árvore é de 6 linhas e a altura máxima da árvore será 24 linhas, incluindo uma linha para a base e duas linhas para o tronco. Em geral o terminal texto tem as seguintes medidas: 80x25 colunasxlinhas.

Linguagem Natural

1. Início
2. Solicite ao usuário um caractere para montar a árvore
3. Solicite ao usuário uma altura ≥ 6 e ≤ 24 ;
4. Caso não seja cumprido as exigências, voltar ao passo 3 até que seja digitado informações dentro dos parâmetros.
5. Montar um algoritmo para montar a árvore
6. Mostrar ao usuário
7. Fim do programa

Linguagem Estruturada

1. Início
2. Declare altura, caractere, copa
3. Leia caractere "Qual o caractere que você quer montar a árvore?"
4. Leia altura
5. Enquanto altura < 6 OU altura > 24 faça
 - 5.1. Leia altura
6. Fim enquanto
7. altura

3) Elabore um algoritmo que receba como parâmetros um vetor de cadeia de caracteres, a quantidade de elementos do vetor e monte na tela o menu correspondente. Ao final acrescente a opção “S Sair”. A função só deverá aceitar os valores entre <1> e o total da quantidade do vetor ou a letra <S>. Qualquer outro caractere digitado deverá ser ignorado.

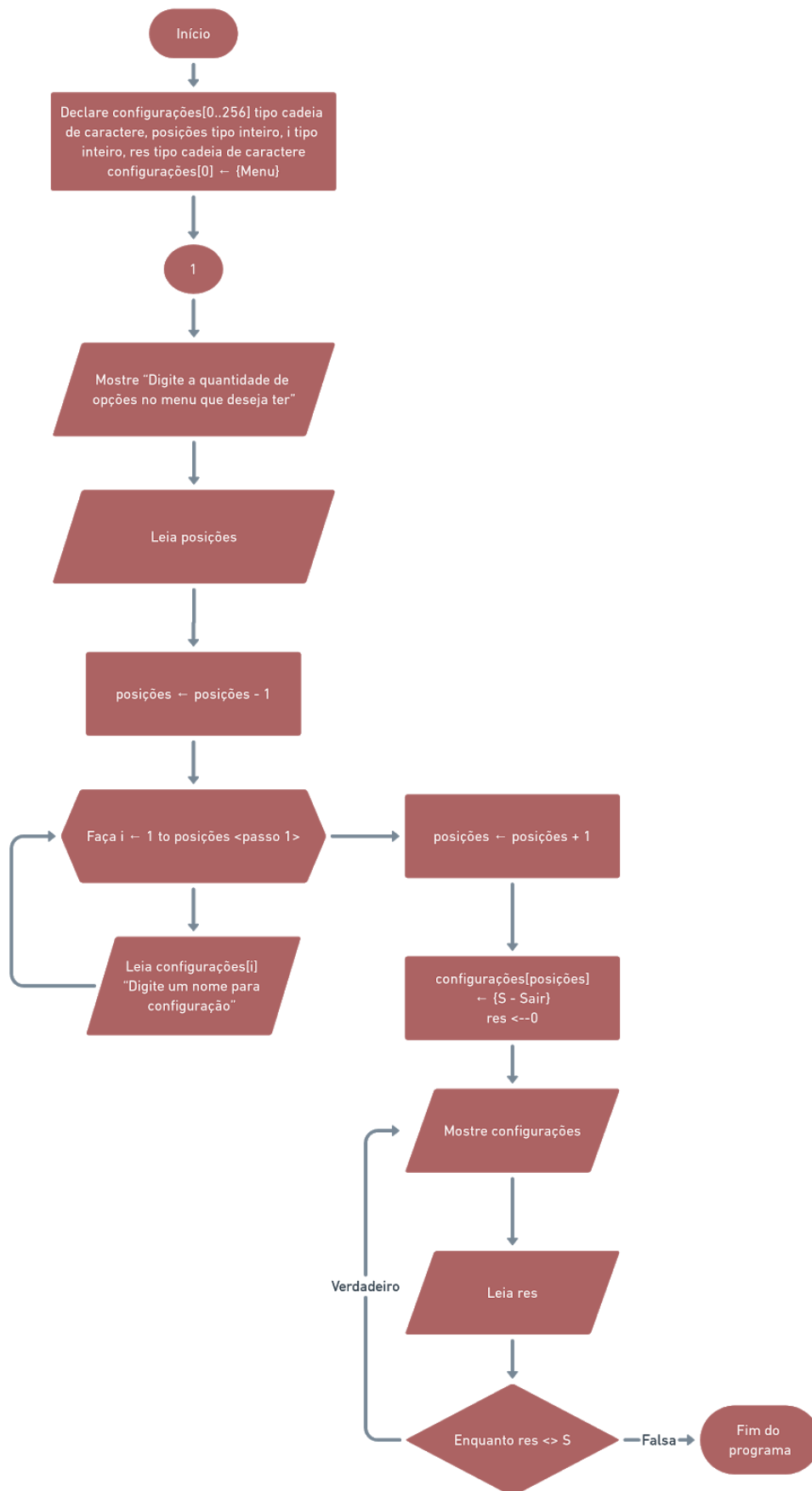
Linguagem Natural

1. Início
2. Solicitar ao usuário a quantidade de opções que terá o menu
3. Ler as posições adicionando-as ao menu
4. Mostrar ao usuário o menu
5. Fim do programa.

Linguagem Estruturada

1. Início
2. Declare configurações[0..256] tipo cadeia de caractere, posições tipo inteiro, i tipo inteiro, res tipo cadeia de caractere
3. Mostre “Digite a quantidade de opções no menu que deseja ter”
4. Leia posições
5. configurações[0] ← {Menu}
6. posições ← posições - 1
7. Faça i ← 1 to posições <passo 1>
 - 7.1. Leia configurações[i] “Digite um nome para configuração”
8. Fim para
9. posições ← posições + 1
10. configurações[posições] ← {S - Sair}
11. Mostre configurações
12. Leia res
13. Enquanto res <> S
 - 13.1. Mostre configurações
 - 13.2. Leia res
14. Fim enquanto
15. Fim do programa

Fluxograma



4) Elabore um algoritmo para calcular a média de consumo de um automóvel. O programa deve receber o total de quilômetros percorridos e o total abastecido e então calcular a média de consumo. Enquanto o usuário não digitar o total de quilômetros negativo o programa continua solicitando os dados para o usuário.

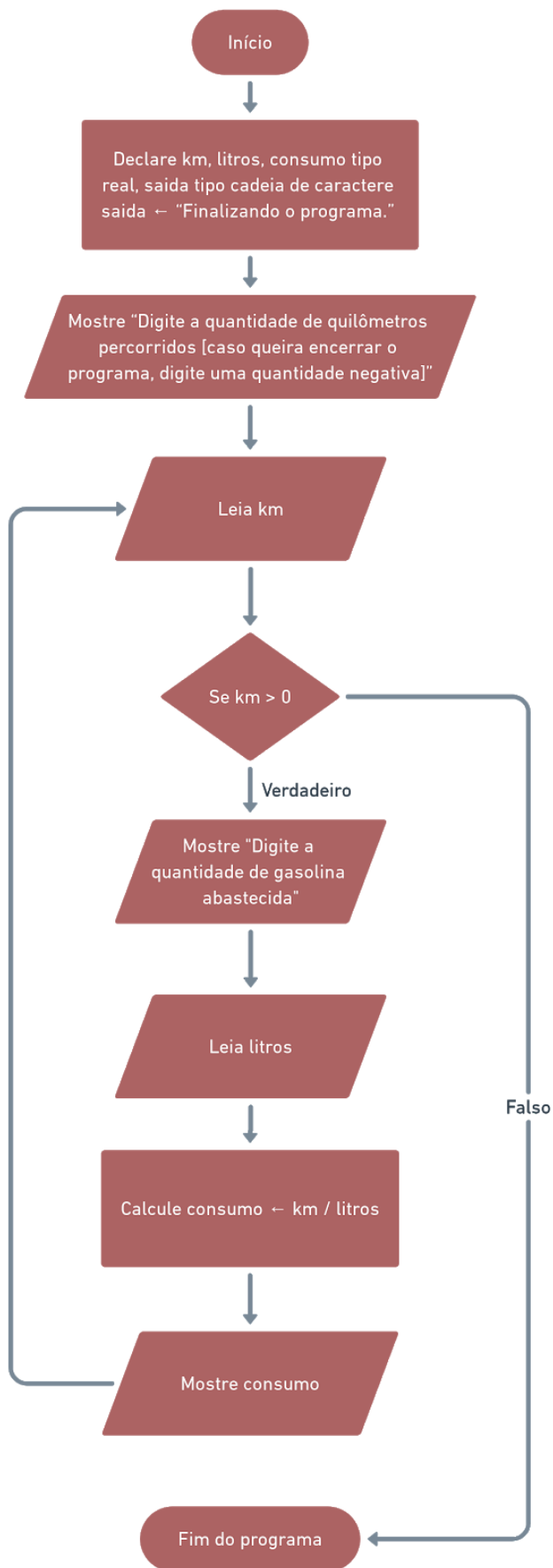
Linguagem Natural

1. Início - consumo do automóvel
2. Solicite ao usuário a quantidade de quilômetros percorridos
 - 2.1. Se a quantidade digitada for < 0 , encerrar o programa
3. Solicite ao usuário o total de gasolina abastecida
4. Calcule a quantidade de quilômetros percorridos sobre o total de gasolina
5. Mostre o consumo ao usuário
6. Faça o passo 2 até que a quantidade de quilômetros seja negativa
7. Fim do programa

Linguagem Estruturada

1. Início
2. Declare km, litros, consumo tipo real, saida tipo cadeia de caractere
3. $saida \leftarrow$ "Finalizando o programa."
4. Mostre "Digite a quantidade de quilômetros percorridos [caso queira encerrar o programa, digite uma quantidade negativa]"
5. Leia km
6. Se $km > 0$, faça
 - 6.1. Mostre "Digite a quantidade de gasolina abastecida."
 - 6.2. Leia litros
 - 6.3. Calcule $consumo \leftarrow km / litros$
7. Se não
 - 7.1. Mostre saida
8. Fim se
9. Fim do programa

Fluxograma



5) Elabore um algoritmo que criptografe uma dada cadeia de caracteres fornecida pelo usuário. Além da cadeia de caracteres, o usuário deverá fornecer o salto que ele quer dar na criptografia. Todo caractere maiúsculo deverá ser transformado em minúsculo antes da criptografia.

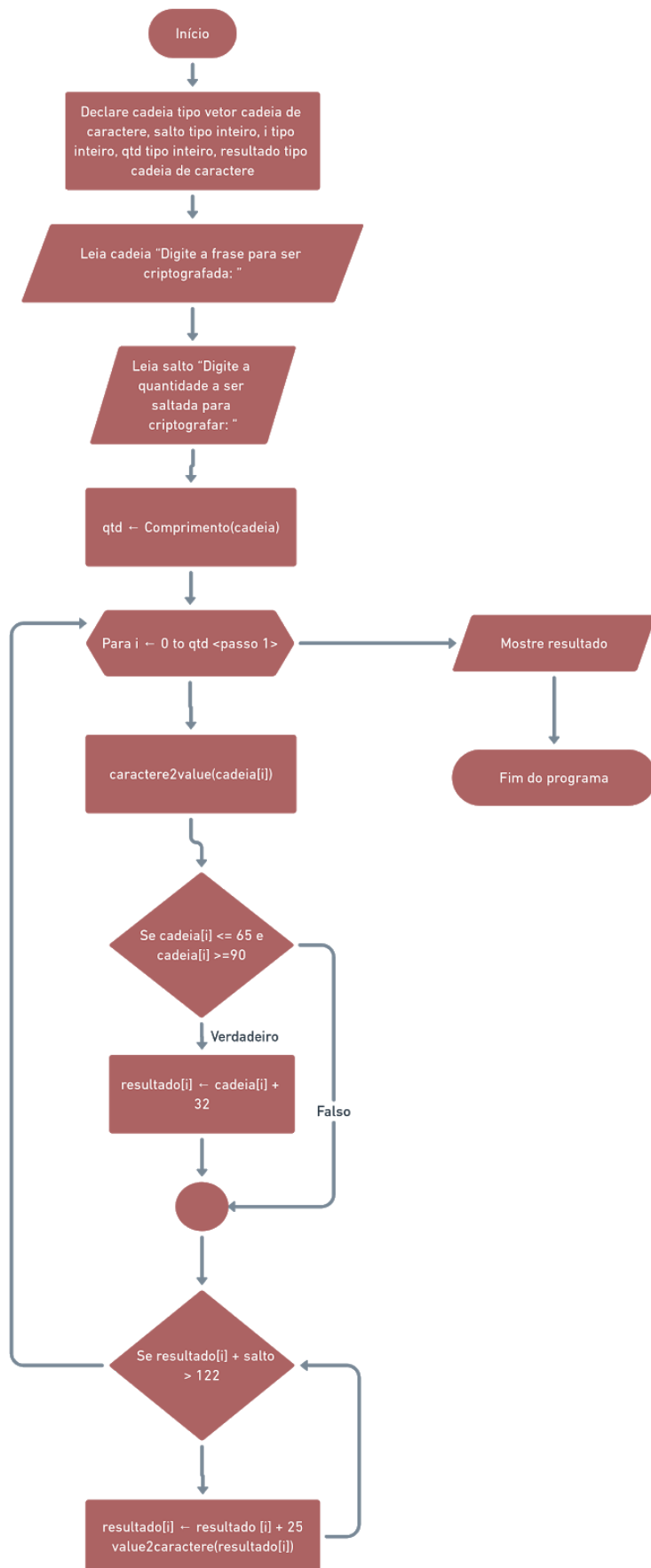
Linguagem Natural

1. Início.
2. Digite uma frase para ser criptografada
3. Digite um número para o salto na criptografia
4. Transforme todas as palavras para minúsculo, se necessário
5. Retorne valor ao usuário
6. Fim do programa

Linguagem Estruturada

1. Início
2. Declare cadeia tipo vetor cadeia de caractere, salto tipo inteiro, i tipo inteiro, qtd tipo inteiro, resultado tipo cadeia de caractere
3. Leia cadeia "Digite a frase para ser criptografada: "
4. Leia salto "Digite a quantidade a ser saltada para criptografar: "
5. $qtd \leftarrow \text{Comprimento}(cadeia)$
6. Para $i \leftarrow 0$ to qtd <passo 1>
 - 6.1. $\text{caractere2value}(cadeia[i])$
 - 6.2. Se $cadeia[i] \leq 65$ e $cadeia[i] \geq 90$
 - 6.2.1. $\text{resultado}[i] \leftarrow cadeia[i] + 32$
 - 6.2.1.1. Se $\text{resultado}[i] + salto > 122$
 - 6.2.1.2. Calcule $\text{resultado}[i] \leftarrow (\text{resultado}[i] + salto) - 122$
 - 6.2.1.3. $\text{resultado}[i] \leftarrow \text{resultado}[i] + 25$
 - 6.2.1.4. $\text{value2caractere}(\text{resultado}[i])$
 - 6.2.1.5. Fim se
 - 6.3. Fim se
 7. Fim para
 8. Mostre resultado
 9. Fim do programa

Fluxograma



6) Elabore um algoritmo que dado um número decimal, transforme em um número binário. A transformação se dá através de divisões sucessivas por 2.

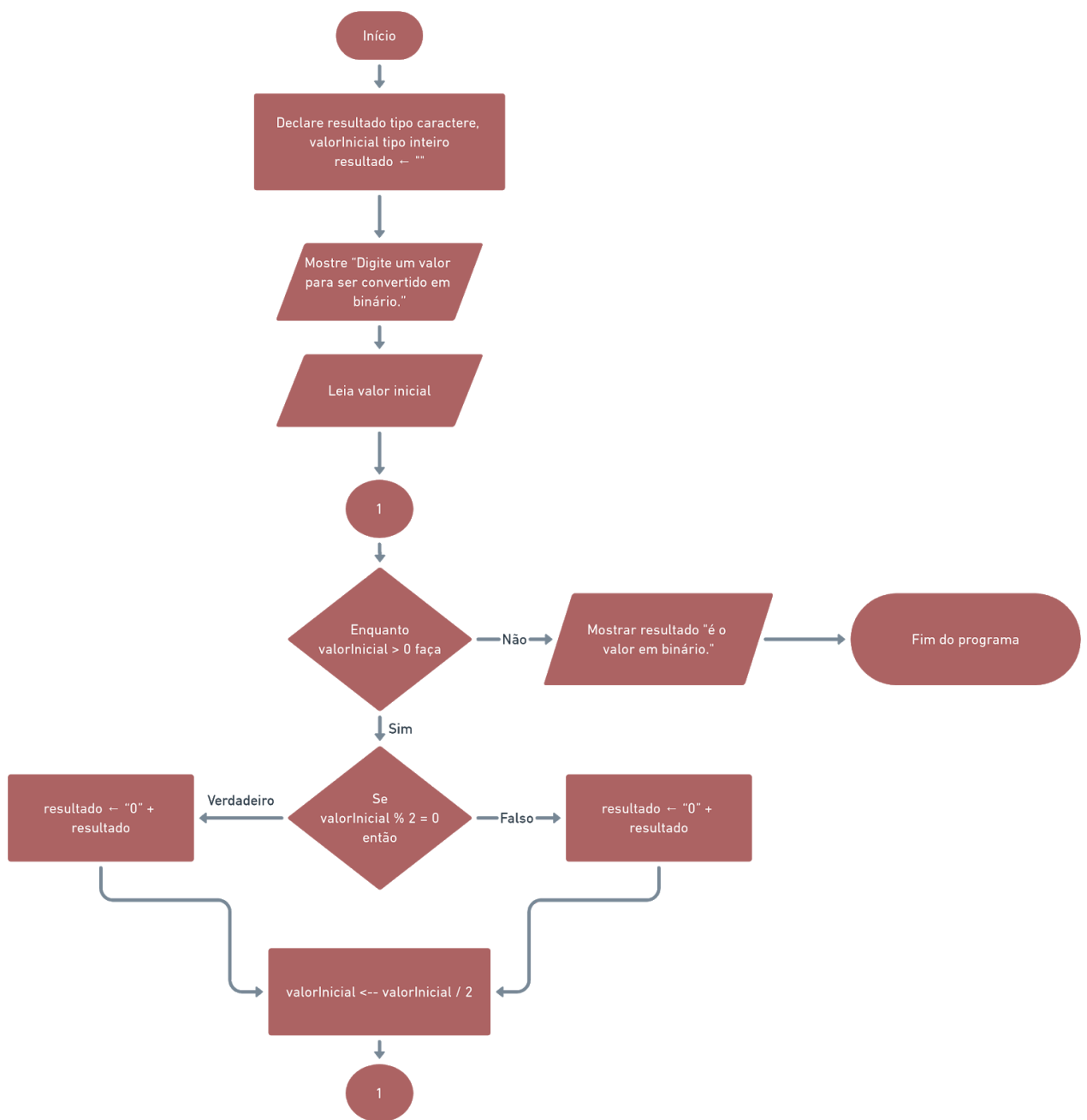
Linguagem Natural

1. Início
2. Solicite ao usuário que insira um número para ser convertido à binário
3. Converter o número em binário
4. Mostrar resultado ao usuário
5. Fim do programa

Linguagem Estruturada

1. Início
2. Declare resultado tipo inteiro, valorInicial tipo inteiro
3. Mostre "Digite um valor para ser convertido em binário."
4. Leia valorInicial
5. resultado \leftarrow ""
6. Enquanto valorInicial > 0 faça
 - 6.1. Se valorInicial % 2 = 0 então
 - 6.1.1. resultado \leftarrow "0" + resultado
 - 6.2. Se não
 - 6.2.1. resultado \leftarrow "1" + resultado
 - 6.3. Fim se
 - 6.4. valorInicial \leftarrow valorInicial / 2
7. Fim enquanto
8. Mostre resultado " é o valor em binário do número digitado."
9. Fim do programa

Fluxograma



7) Elabore um algoritmo que dado um valor binário, transforme para a representação decimal.

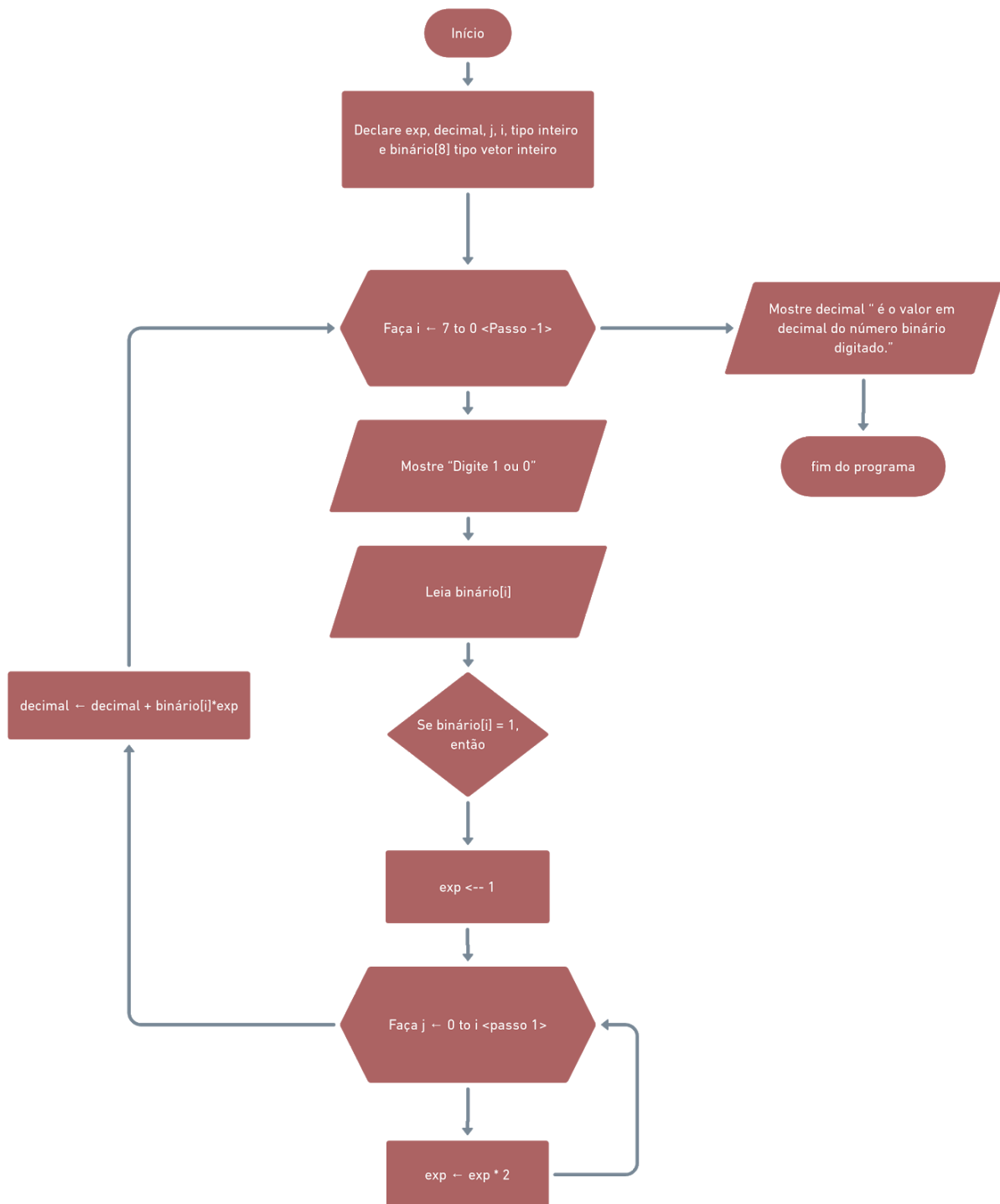
Linguagem Natural

1. Início
2. Solicite ao usuário que insira um número binário para ser convertido a decimal
3. Converter o número para decimal
4. Mostrar resultado ao usuário
5. Fim do programa

Linguagem Estruturada

1. Início
2. Declare exp, decimal, j, i, tipo inteiro e binário[8] tipo vetor inteiro
3. Faça $i \leftarrow 7$ to 0 <Passo -1>
 - 3.1. Mostre “Digite 1 ou 0”
 - 3.2. Leia binário[i]
 - 3.3. Se binário[i] = 1, então
 - 3.3.1. $exp \leftarrow 1$
 - 3.3.2. Faça $j \leftarrow 0$ to i <passo 1>
 - 3.3.2.1. $exp \leftarrow exp * 2$
 - 3.3.3. Fim faça
 - 3.3.4. $decimal \leftarrow decimal + binário[i] * exp$
 - 3.4. Fim se
4. Fim faça
5. Mostre decimal “ é o valor em decimal do número binário digitado.”
6. Fim do programa

Fluxograma



8) Elabore um algoritmo que some dois vetores de números inteiros, colocando o resultado em um terceiro vetor. Os vetores deverão ter o mesmo comprimento, se não a soma não poderá ser realizada.

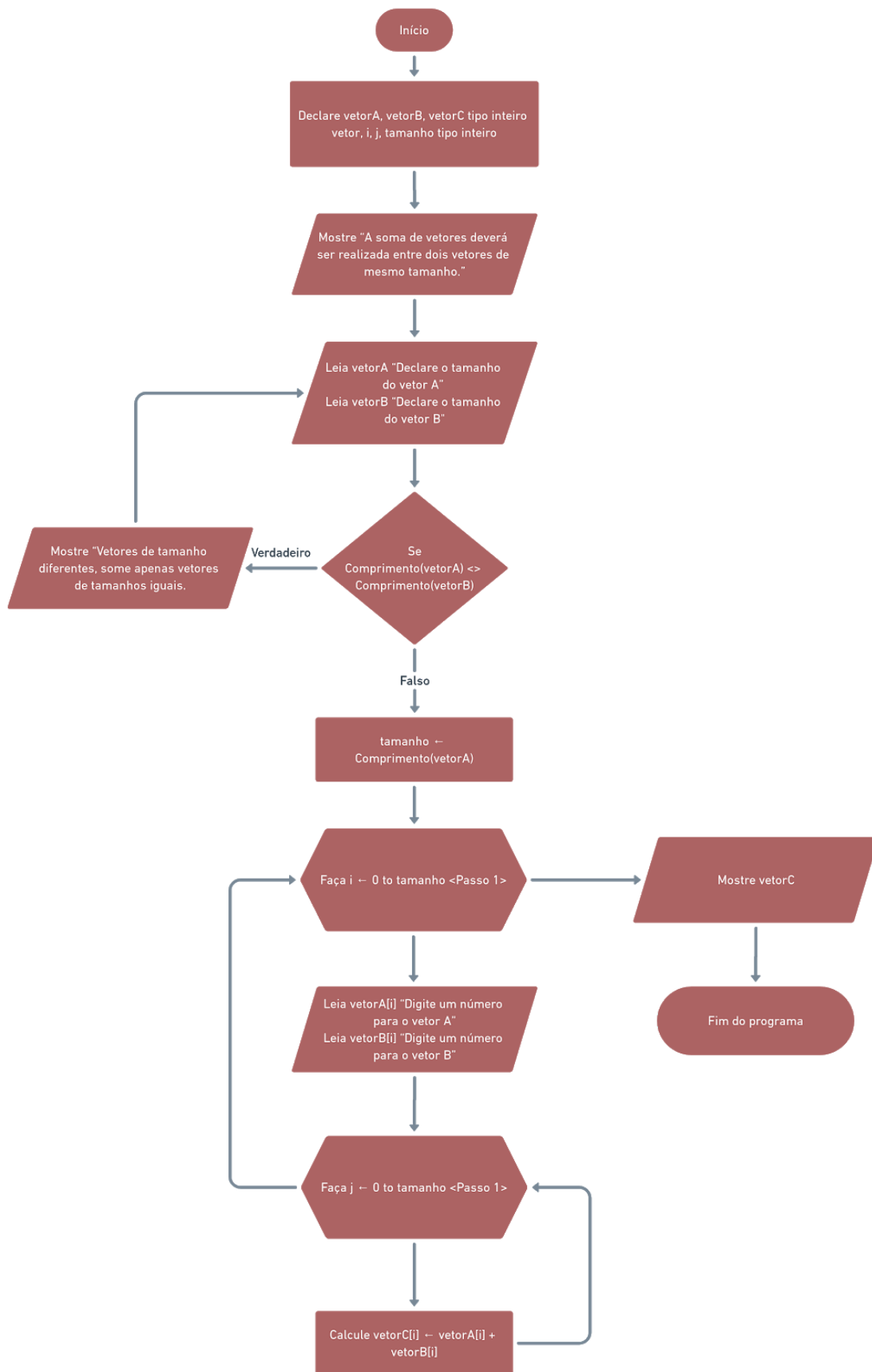
Linguagem Natural

1. Início
2. Solicite ao usuário o tamanho dos vetores A e B
3. Verifique se o tamanho dos dois é o mesmo.
4. Se sim, solicite a uma quantidade de números correspondentes para A e B
5. Some os valores
6. Retorne ao usuário
7. Fim do programa

Linguagem Estruturada

1. Início
2. Declare vetorA, vetorB, vetorC tipo inteiro vetor, i
3. Mostre “A soma de vetores deverá ser realizada entre dois vetores de mesmo tamanho.”
4. Leia vetorA “Declare o tamanho do vetor A”
5. Leia vetorB “Declare o tamanho do vetor B”
6. Se Comprimento(vetorA) \neq Comprimento(vetorB)
 - 6.1. Mostre “Vetores de tamanho diferentes, some apenas vetores de tamanhos iguais.
7. Se não
 - 7.1. $i \leftarrow \text{Comprimento}(\text{vetorA})$
 - 7.2. Faça $i \leftarrow 0$ to Comprimento(vetorA) <Passo 1>
 - 7.2.1. Leia vetorA[i] “Digite um número para o vetor A”
 - 7.2.2. Leia vetorB[i] “Digite um número para o vetor B”
 - 7.3. Faça $j \leftarrow 0$ to Comprimento(vetorB) <Passo 1>
 - 7.3.1. Calcule $\text{vetorC}[i] \leftarrow \text{vetorA}[i] + \text{vetorB}[j]$
 - 7.4. Fim faça
8. Fim se
9. Mostre vetorC
10. Fim do programa

Fluxograma



9) Elabore um algoritmo para que seja possível dar um reajuste aos funcionários de uma empresa de acordo com a tabela abaixo:

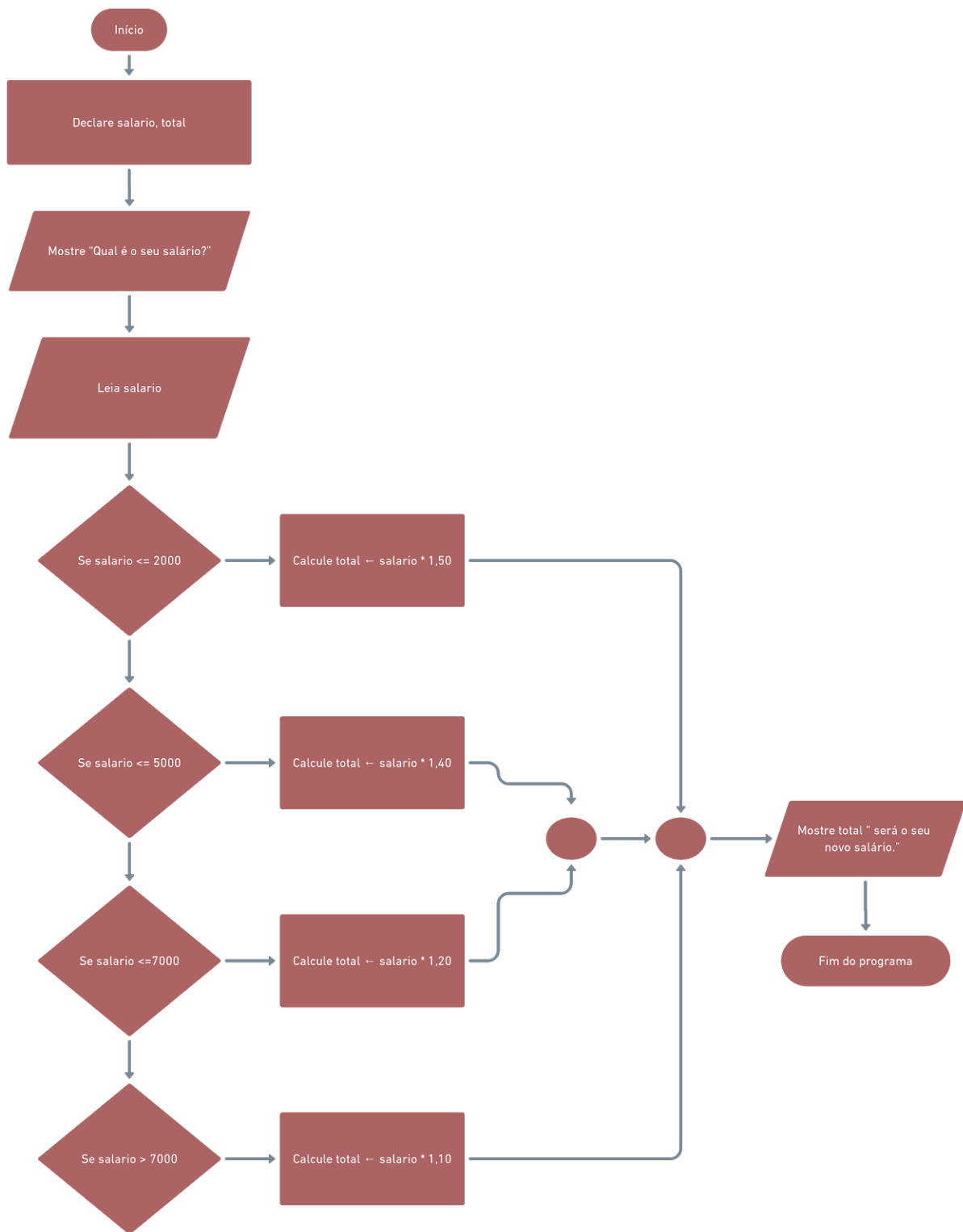
Linguagem Natural

1. Início
2. Solicite ao usuário o valor do salário para saber o reajuste
3. Calcular o reajuste com base na tabela
4. Retornar o valor ao usuário
5. Fim do programa

Linguagem Estruturada

1. Início
2. Declare salario, total
3. Mostre “Qual é o seu salário?”
4. Leia salario
5. Se salario \leq 2000
 - 5.1. Calcule total \leftarrow salario * 1,50
6. Se salario \leq 5000
 - 6.1. Calcule total \leftarrow salario * 1,40
7. Se salario \leq 7000
 - 7.1. Calcule total \leftarrow salario * 1,20
8. Se salario $>$ 7000
 - 8.1. Calcule total \leftarrow salario * 1,10
9. Fim se
10. Mostre total “ será o seu novo salário.”
11. Fim do programa

Fluxograma



10) Elabore um algoritmo que realizar o pedido de uma lanchonete, o usuário deverá digitar o código e a quantidade. Enquanto a quantidade for um número positivo, o programa deverá continuar solicitando o código para um novo item. Quando o usuário digitar um número negativo no código do produto, o programa deverá mostrar a lista de pedidos com o total a ser pago. A tabela abaixo tem os códigos, nome do produto e o valor.

Linguagem Natural

1. Início
2. Solicite ao usuário o código do produto
3. Solicite a quantidade que deseja comprar desse produto
4. Somar ao total esse valor
5. Repetir passo 2 até que o valor digitado seja negativo
6. Mostrar ao usuário a lista e o total a pagar
7. Fim do programa

Linguagem Estruturada

1. Início
2. Declare código[4] tipo vetor inteiro, valores[4] tipo vetor real, total
3. valores[4] \leftarrow {4.50, 5.00, 2.00, 6.00}
4. Leia res "Digite o código do produto: "
5. Leia qntd "Digite a quantidade do produto"
- 5.1. Caso res ≥ 100 e res ≤ 103
 - 5.1.1. Se res = 100
 - 5.1.1.1. código[0] \leftarrow {código[0] + res}
 - 5.1.2. Se res = 101
 - 5.1.2.1. código[1] \leftarrow {código[1] + res}
 - 5.1.3. Se res = 102
 - 5.1.3.1. código[2] \leftarrow {código[2] + res}
 - 5.1.4. Se res = 103
 - 5.1.4.1. código[3] \leftarrow {código[3] + res}
 - 5.1.5. Fim se
- 5.2. Fim caso
6. Se res < 0
 - 6.1. Calcule total \leftarrow código[0..4]*valores[0..4]
7. Fim se
8. Mostre código
9. Mostre total
10. Fim do programa

Fluxograma

