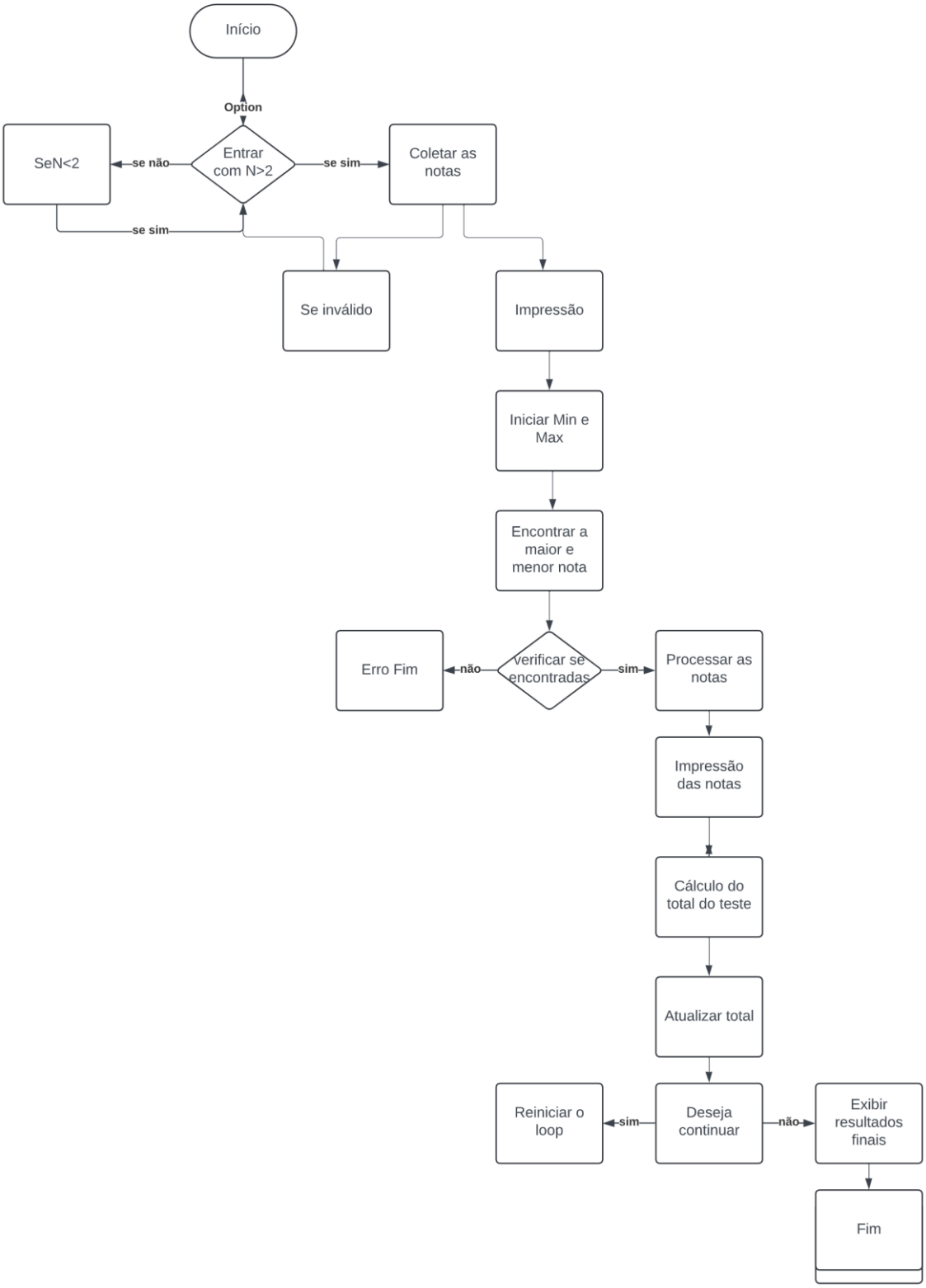
Disciplina: Programação Estruturada e Modular -> prof. Carlos Veríssimo.

Objetivo: Programa para cálculo de notas.

Data: 23/08/2024

Autor: Cyntia Farias Ruffo


Início
Option
Entrar com N>2
se não
SeN<2
se sim
Coletar as notas
se sim
Se inválido
Impressão
Iniciar Min e Max
Encontrar a maior e menor nota
verificar se encontradas
não
Erro Fim
sim
Processar as notas
Impressão das notas
Cálculo do total do teste
Atualizar total
Deseja continuar
sim
Reiniciar o loop
não
Exibir resultados finais
Fim

```c
#include <stdio.h>


#define MAX_NOTAS 100
#define MAX_TESTES 100

int main() {
    int N;
    float notas[MAX_NOTAS];
    float nota;
    int i;
    float min, max;
    int min_index, max_index;
    int continuar;
```

```
float total_teste;
float totais_testes[MAX_TESTES]; // Array para armazenar os totais dos testes
int num_testes = 0;
float total_geral = 0;

do {
   // Inicializa vari□veis para o novo teste
   total_teste = 0;
   min_index = max_index = -1;

   // Solicita o valor de N e garante que seja maior que 2
   do {
      printf("\nInsira o valor de N: (OBS: N > 2) ");
      scanf("%d", &N);

      if (N <= 2) {
         printf("INSIRA NOVAMENTE! O VALOR DE N DEVE SER > 2!\n");
      }
   } while (N <= 2);

   // Coleta as notas
   for (i = 0; i < N; i++) {
      while (1) { // Loop para garantir que a nota esteja entre 0 e 10
         printf("Insira a %d□ nota: ", i + 1);
         scanf("%f", &nota);
```

```
        if (nota >= 0 && nota <= 10) {
            break; // Nota v□lida, sai do loop
        } else {
            printf("NOTA INV□LIDA! INSIRA NOVAMENTE!\n");
        }
    }
    notas[i] = nota;
}

// Imprime as notas
printf("\nNotas inseridas:\n");
for (i = 0; i < N; i++) {
    printf("%.2f ", notas[i]);
}
printf("\n");

// Inicializa min e max com o primeiro valor do array
if (N > 0) {
    min = max = notas[0];
    min_index = max_index = 0;

    // Encontra a menor e a maior nota
    for (i = 1; i < N; i++) {
        if (notas[i] < min) {
            min = notas[i];
            min_index = i;
```

```
    }
    if (notas[i] > max) {
        max = notas[i];
        max_index = i;
    }
}

// Verifica se a menor e maior nota foram encontrados
if (min_index == -1 || max_index == -1) {
    printf("Erro: N□o foi poss□vel encontrar a menor ou a maior nota.\n");
    return 1; // C□digo de erro
}

// Cria um novo array para armazenar as notas restantes
float notas_restantes[MAX_NOTAS];
int j = 0;

// Copia as notas, excluindo a menor e a maior
for (i = 0; i < N; i++) {
    if (i != min_index && i != max_index) {
        notas_restantes[j++] = notas[i];
    }
}

// Atualiza o n□mero de notas restantes
int N_restantes = j;
```

```
        // Imprime as notas restantes
        printf("\nNotas restantes (sem a menor e a maior nota):\n");
        for (i = 0; i < N_restantes; i++) {
            printf("%.2f ", notas_restantes[i]);
        }
        printf("\n");

        // Calcula o total do teste
        for (i = 0; i < N_restantes; i++) {
            total_teste += notas_restantes[i]; // Adiciona as notas restantes ao total do teste
        }

        printf("\nTotal do Teste: %.2f", total_teste);

        // Atualiza o total geral e armazena o total do teste
        total_geral += total_teste;
        totais_testes[num_testes] = total_teste;
        num_testes++;

        // Pergunta ao usu□rio se deseja continuar
        printf("\n\nDeseja realizar outro teste? (1 - Sim, 0 - N□o): ");
        scanf("%d", &continuar);


    } else {
```

```
            printf("Erro: O array de notas est vazio.\n");
            return 1; // C digo de erro
        }
    } while (continuar);



    // Exibe os totais finais quando o usu rio decide parar
    for (i = 0; i < num_testes; i++) {
        printf("Final do teste %d: %.2f\n", i + 1, totais_testes[i]);
    }
    printf("Total Geral: %.2f\n", total_geral);
    printf("N mero de Testes: %d\n", num_testes);

    return 0;
}
```