Nome: Messias Feres Curi Melo
Matrícula: 2022003764

1 Tipos Abstratos de Dados

1.1 TAD é uma **estrutura de dados** que possui detalhes 'escondidos' durante a implementação do software, facilitando para se programar, reutilizar códigos, fazer manutenção, para segurança do código, entre outras vantagens. No geral ele faz uma abstração dos dados e modulariza o projeto.

1.2

main.c

C: > Users > Messi > OneDrive > Área de Trabalho > lista3 > C main.c > main()

```c
#include "cubo.h"
#include <stdio.h>
#include <stdlib.h>

int main(){
    int n;
    Cubo c;
    printf("Número de lados : ");
    scanf("%d", &n);
    criarCubo(&c, n);
    printf("-> Valor do lado : %d\n", medidaCubo(c.lado));
    printf("-> Valor da área : %d\n", areaCubo(c.lado));
    printf("-> Valor do volume : %d\n", volumeCubo(c.lado));
    return 0;
}
```

cubo.c

C: > Users > Messi > OneDrive > Área de Trabalho > lista3 > C cubo.c > medidaCubo(int)

```c
#include "cubo.h"
#include <stdio.h>
#include <stdlib.h>

void criarCubo(Cubo *c, int n){
    c->lado = n;
}

int medidaCubo(int n){
    return n;
}

int areaCubo(int n){
    return (n*n)*6;
}

int volumeCubo(int n){
    return (n*n*n);
}
```

cubo.h

C: > Users > Messi > OneDrive > Área de Trabalho > lista3 > C cubo.h > areaCubo(int)

```c
#ifndef CUBO_H
#define CUBO_H

typedef struct{
    int lado;
}Cubo;

void criarCubo(Cubo *c, int n);
int medidaCubo(int n);
int areaCubo(int n);
int volumeCubo(int n);

#endif
```

Terminal:

```
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/lista3$ gcc main.c cubo.c cubo.h -o tp2
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/lista3$ ./tp2
Número de lados : 10
-> Valor do lado : 10
-> Valor da área : 600
-> Valor do volume : 1000
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/lista3$
```

1.3
main.c

C: > Users > Messi > OneDrive > Área de Trabalho > lista3_3 > C main.c > main()

```c
#include "conjunto.h"
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

int main(){
    int escolha, elemento, conjuntoNumero, conjuntoNumero2, resultado;
    listadeConjuntos c;
    c.tamanho = 0;
    c.conjuntos = NULL;

    printf("--- MENU ---\n");
    printf("1 - Criar um conjunto vazio\n");
    printf("2 - União de dois conjuntos\n");
    printf("3 - Inserir elemento\n");
    printf("4 - Remover elemento\n");
    printf("5 - Intersecção entre dois conjuntos\n");
    printf("6 - Diferença de dois conjuntos\n");
    printf("7 - Número pertence ao conjunto\n");
    printf("8 - Menor valor do conjunto\n");
    printf("9 - Maior valor do conjunto\n");
    printf("10 - Testar se dois conjuntos são iguais\n");
    printf("11 - Tamanho do conjunto\n");
    printf("12 - Testar se o conjunto é vazio\n");
    printf("0 - Finalizar\n");

    while (true) {
        printf("Valor de escolha : ");
        scanf("%d", &escolha);

        switch (escolha) {
        case 1:
            criarConjunto(&c);
            printf("Novo conjunto criado\n");
            break;
        case 2:
            printf("Insira o número do primeiro conjunto : ");
            scanf("%d", &conjuntoNumero);
            printf("Insira o número do segundo conjunto : ");
            scanf("%d", &conjuntoNumero2);
            Conjunto uniao = uniaoConjuntos(c.conjuntos[conjuntoNumero - 1], c.conjuntos[conjuntoNumero2 - 1]);
            printf("União dos conjuntos é : ");
            for(int i = 0; i<uniao.quantidadeElementos; i++){
                printf("%d ", uniao.elementos[i]);
            }
            limpar(&uniao);
            printf("\n");
            break;
        case 3:
            printf("Insira qual o número do conjunto : ");
            scanf("%d", &conjuntoNumero);
            printf("Insira o número a ser adicionado : ");
            scanf("%d", &elemento);
            inserirElemento(&c.conjuntos[conjuntoNumero-1], elemento);
            break;
        case 4:
            printf("Insira qual o número do conjunto : ");
            scanf("%d", &conjuntoNumero);
            printf("Insira o número a ser removido : ");
            scanf("%d", &elemento);
            removerElemento(&c.conjuntos[conjuntoNumero-1], elemento);
            break;
        case 5:
            printf("Insira o número do primeiro conjunto : ");
            scanf("%d", &conjuntoNumero);
            printf("Insira o número do segundo conjunto : ");
            scanf("%d", &conjuntoNumero2);
            Conjunto intersecao = intersecaoConjuntos(c.conjuntos[conjuntoNumero - 1], c.conjuntos[conjuntoNumero2 - 1]);
            printf("Intersecção dos conjuntos é : ");
            for(int i = 0; i<intersecao.quantidadeElementos; i++){
                printf("%d ", intersecao.elementos[i]);
            }
            limpar(&intersecao);
            printf("\n");
            break;
        case 6:
```

```c
            printf("Insira o número do primeiro conjunto : ");
            scanf("%d", &conjuntoNumero);
            printf("Insira o número do segundo conjunto : ");
            scanf("%d", &conjuntoNumero2);
            Conjunto diferenca = diferencaConjuntos(c.conjuntos[conjuntoNumero - 1], c.conjuntos[conjuntoNumero2 - 1]);
            printf("Diferença dos conjuntos é : ");
            for(int i = 0; i<diferenca.quantidadeElementos; i++){
                printf("%d ", diferenca.elementos[i]);
            }
            limpar(&diferenca);
            printf("\n");
            break;
        case 7:
            printf("Insira qual o número do conjunto : ");
            scanf("%d", &conjuntoNumero);
            printf("Insira o número a ser verificado : ");
            scanf("%d", &elemento);
            resultado = pertence(c.conjuntos[conjuntoNumero-1], elemento);
            if(resultado){
                printf("Pertence\n");
            }else{
                printf("Não pertence\n");
            }
            break;
        case 8:
            printf("Insira qual o número do conjunto : ");
            scanf("%d", &conjuntoNumero);
            resultado = menorValor(c.conjuntos[conjuntoNumero-1]);
            printf("O menor elemeno é : %d\n", resultado);
            break;
        case 9:
            printf("Insira qual o número do conjunto : ");
            scanf("%d", &conjuntoNumero);
            resultado = maiorValor(c.conjuntos[conjuntoNumero-1]);
            printf("O maior elemeno é : %d\n", resultado);
            break;
        case 10:
            printf("Insira o número do primeiro conjunto : ");
            scanf("%d", &conjuntoNumero);
            printf("Insira o número do segundo conjunto : ");
            scanf("%d", &conjuntoNumero2);
            resultado = saoIguais(c.conjuntos[conjuntoNumero-1], c.conjuntos[conjuntoNumero2-1]);
            if(resultado){
                printf("São iguais\n");
            }else{
                printf("Não são iguais\n");
            }
            break;
        case 11:
            printf("Insira qual o número do conjunto : ");
            scanf("%d", &conjuntoNumero);
            resultado = tamanho(c.conjuntos[conjuntoNumero-1]);
            printf("O tamanho é : %d\n", resultado);
            break;
        case 12:
            printf("Insira qual o número do conjunto : ");
            scanf("%d", &conjuntoNumero);
            resultado = eVazio(c.conjuntos[conjuntoNumero-1]);
            if(resultado){
                printf("É vazio\n");
            }else{
                printf("Não é vazio\n");
            }
            break;
        case 0:
            break;
        default:
            printf("Escolha inválida\n");
            break;
        }
        if(escolha == 0){
            for(int i = 0; i < c.tamanho; i++){
                limpar(&c.conjuntos[i]);
            }
            break;
        }
    }

    return 0;
}
```

conjunto.c

C: > Users > Messi > OneDrive > Área de Trabalho > lista3_3 > C conjunto.c > criarConjunto(listadeConjuntos *)

```c
#include "conjunto.h"
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

Conjunto criar_conjunto_vazio() {
    Conjunto n;
    n.elementos = NULL;
    n.quantidadeElementos = 0;
    return n;
}

void criarConjunto(listadeConjuntos *c){
    if(c->tamanho == 0) {
        c->tamanho = 1;
        c->conjuntos = (Conjunto*)malloc(sizeof(Conjunto));
        c->conjuntos[0] = criar_conjunto_vazio();
    }else {
        c->tamanho++;
        int nt = sizeof(listadeConjuntos);
        c->conjuntos = (Conjunto*)realloc(c->conjuntos, c->tamanho*sizeof(Conjunto));
        c->conjuntos[c->tamanho-1] = criar_conjunto_vazio();
    }
}

Conjunto uniaoConjuntos(Conjunto c1, Conjunto c2) {
    Conjunto uniao = criar_conjunto_vazio();
    for (int i = 0; i < c1.quantidadeElementos; i++) {
        inserirElemento(&uniao, c1.elementos[i]);
    }
    for (int i = 0; i < c2.quantidadeElementos; i++) {
        if (!pertence(uniao, c2.elementos[i])) {
            inserirElemento(&uniao, c2.elementos[i]);
        }
    }
    return uniao;
}

void inserirElemento(Conjunto *c, int elemento) {
    for (int i = 0; i < c->quantidadeElementos; i++) {
        if (c->elementos[i] == elemento) {
            return;
        }
    }
    c->quantidadeElementos++;
    c->elementos = (int *)realloc(c->elementos, c->quantidadeElementos * sizeof(int));
    c->elementos[c->quantidadeElementos - 1] = elemento;
}

void removerElemento(Conjunto *c, int elemento) {
    for (int i = 0; i < c->quantidadeElementos; i++) {
        if (c->elementos[i] == elemento) {
            for (int j = i; j < c->quantidadeElementos - 1; j++) {
                c->elementos[j] = c->elementos[j + 1];
            }
            c->quantidadeElementos--;
            c->elementos = (int *)realloc(c->elementos, c->quantidadeElementos * sizeof(int));
            return;
        }
    }
}

Conjunto intersecaoConjuntos(Conjunto c1, Conjunto c2) {
```

```
    Conjunto intersecao = criar_conjunto_vazio();
    for (int i = 0; i < c1.quantidadeElementos; i++) {
        if (pertence(c2, c1.elementos[i])) {
            inserirElemento(&intersecao, c1.elementos[i]);
        }
    }
    return intersecao;
}

Conjunto diferencaConjuntos(Conjunto c1, Conjunto c2) {
    Conjunto diferenca = criar_conjunto_vazio();
    for (int i = 0; i < c1.quantidadeElementos; i++) {
        if (!pertence(c2, c1.elementos[i])) {
            inserirElemento(&diferenca, c1.elementos[i]);
        }
    }
    return diferenca;
}

int pertence(Conjunto c, int numero) {
    for (int i = 0; i < c.quantidadeElementos; i++) {
        if (c.elementos[i] == numero) {
            return 1;
        }
    }
    return 0;
}

int menorValor(Conjunto c) {
    if (c.quantidadeElementos == 0) {
        return -1;
    }
    int menor = c.elementos[0];
    for (int i = 1; i < c.quantidadeElementos; i++) {
        if (c.elementos[i] < menor) {
            menor = c.elementos[i];
        }
    }
    return menor;
}

int maiorValor(Conjunto c) {
    if (c.quantidadeElementos == 0) {
        return -1;
    }
    int maior = c.elementos[0];
    for (int i = 1; i < c.quantidadeElementos; i++) {
        if (c.elementos[i] > maior) {
            maior = c.elementos[i];
        }
    }
    return maior;
}

int saoIguais(Conjunto c1, Conjunto c2) {
    if (c1.quantidadeElementos != c2.quantidadeElementos) {
        return false;
    }
```

```
    for (int i = 0; i < c1.quantidadeElementos; i++) {
        if (!pertence(c2, c1.elementos[i])) {
            return false;
        }
    }
    return true;
}

int tamanho(Conjunto c) {
    if(c.elementos == NULL || c.quantidadeElementos == 0){
        return 0;
    }else{
        return c.quantidadeElementos;
    }
}

int eVazio(Conjunto c) {
    if(c.elementos == NULL || c.quantidadeElementos == 0){
        return true;
    }else{
        return false;
    }
}

void limpar(Conjunto *c) {
    free(c->elementos);
    c->elementos = NULL;
    c->quantidadeElementos = 0;
}
```

conjunto.h

C: > Users > Messi > OneDrive > Área de Trabalho > lista3_3 > C conjunto.h > criar_conjunto_vazio()

```
#ifndef CONJUNTO_H
#define CONJUNTO_H

typedef struct{
    int *elementos;
    int quantidadeElementos;
}Conjunto;

typedef struct{
    Conjunto *conjuntos;
    int tamanho;
}listadeConjuntos;

Conjunto criar_conjunto_vazio();
void criarConjunto(listadeConjuntos *c);
Conjunto uniaoConjuntos(Conjunto c1, Conjunto c2);
void inserirElemento(Conjunto *c, int elemento);
void removerElemento(Conjunto *c, int elemento);
Conjunto intersecaoConjuntos(Conjunto c1, Conjunto c2);
Conjunto diferencaConjuntos(Conjunto c1, Conjunto c2);
int pertence(Conjunto c, int numero);
int menorValor(Conjunto c);
int maiorValor(Conjunto c);
int saoIguais(Conjunto c1, Conjunto c2);
int tamanho(Conjunto c);
int eVazio(Conjunto c);
void limpar(Conjunto *c);

#endif
```

Terminal:

```
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/lista3_3$ gcc main.c conjunto.c conjunto.h -o tp3
^[[Amessiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/lista3_3$ ./tp3
--- MENU ---
1 - Criar um conjunto vazio
2 - União de dois conjuntos
3 - Inserir elemento
4 - Remover elemento
5 - Intersecção entre dois conjuntos
6 - Diferença de dois conjuntos
7 - Número pertence ao conjunto
8 - Menor valor do conjunto
9 - Maior valor do conjunto
10 - Testar se dois conjuntos são iguais
11 - Tamanho do conjunto
12 - Testar se o conjunto é vazio
0 - Finalizar
Valor de escolha : 1
Novo conjunto criado
Valor de escolha : 1
Novo conjunto criado
Valor de escolha : 3
Insira qual o número do conjunto : 1
Insira o número a ser adicionado : 2
Valor de escolha : 3
Insira qual o número do conjunto : 2
Insira o número a ser adicionado : 1
Valor de escolha : 3
Insira qual o número do conjunto : 1
Insira o número a ser adicionado : 23
Valor de escolha : 8
Insira qual o número do conjunto : 1
O menor elemeno é : 2
Valor de escolha : 6
Insira o número do primeiro conjunto : 1
Insira o número do segundo conjunto : 2
Diferença dos conjuntos é : 2 23
Valor de escolha : 11
Insira qual o número do conjunto : 1
O tamanho é : 2
Valor de escolha : 12
Insira qual o número do conjunto : 2
Não é vazio
Valor de escolha : 7
Insira qual o número do conjunto : 1
Insira o número a ser verificado : 2
Pertence
Valor de escolha : 2
Insira o número do primeiro conjunto : 1
Insira o número do segundo conjunto : 2
União dos conjuntos é : 2 23 1
Valor de escolha : 0
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/lista3_3$
```