

CÓDIGO PARA CRIAR GRAFOS

```
#include <stdio.h>
#include <stdlib.h>

struct grafos{
    int ehPonderado;
    int nVertices;
    int ** arestas;
    float ** pesos;
    int * grau;
};

typedef struct grafos Grafo;

Grafo* criaGrafo(int Vertices, int ehPonderado){
    Grafo *gr;

    gr = (Grafo *) malloc(sizeof(struct grafos));
    if (gr != NULL){
        gr->nVertices = Vertices;
        gr->ehPonderado = (ehPonderado != 0)?1:0;
        gr->grau = (int *) calloc(Vertices,sizeof(int));
        gr->arestas = (int **) malloc(Vertices*sizeof(int*));
        for(int i=0;i < Vertices;i++){
            gr->arestas[i] = (int *) malloc(Vertices*sizeof(int));
        }

        if(gr->ehPonderado){
            gr->pesos = (float **) malloc(Vertices*sizeof(float*));
            for(int i=0;i < Vertices;i++){
                gr->pesos[i] = (float *) malloc(Vertices*sizeof(float));
            }
        }
    }

    return gr;
}

void liberaGrafo(Grafo* gr){
    if(gr != NULL){
        int i;
        for(int i=0;i < gr->nVertices;i++){
            free(gr->arestas[i]);
        }
        free(gr->arestas);

        if(gr->ehPonderado){
            for(int i=0;i < gr->nVertices;i++){
                free(gr->pesos[i]);
            }
            free(gr->pesos);
        }
        free(gr->grau);
        free(gr);
    }
}
```

```

int insereAresta(Grafo* gr,int origem,int destino,int digrafo,float peso){
    int criou = 0;
    if(gr != NULL){
        if(origem >= 0 && origem < gr->nVertices){
            if(destino >= 0 && destino < gr->nVertices){
                gr->arestas[origem][gr->grau[origem]] = destino;
                if(gr->ehPonderado)
                    gr->pesos[origem][gr->grau[origem]] = peso;
                (gr->grau[origem])++;

                if (digrafo == 0)
                    insereAresta(gr,destino,origem,1,peso);
                criou = 1;
            }
        }
    }
    return criou;
}

```

```

void mostraGrafo(Grafo *gr){
    printf("origem \t\t destino \n");
    for(int i=0; i < gr->nVertices;i++)
        for(int d=0; d < gr->grau[i];d++)
            if(gr->arestas[i][d] != 0)
                printf("%d \t\t %d \n",i,gr->arestas[i][d]);
}

```

```

int main() {
    int ehDigrafo = 1;
    Grafo *gr;

    gr = criaGrafo(5,0);

    insereAresta(gr,0,1,ehDigrafo,0);
    insereAresta(gr,1,3,ehDigrafo,0);
    insereAresta(gr,1,2,ehDigrafo,0);
    insereAresta(gr,2,4,ehDigrafo,0);
    insereAresta(gr,3,0,ehDigrafo,0);
    insereAresta(gr,3,4,ehDigrafo,0);
    insereAresta(gr,4,1,ehDigrafo,0);

    mostraGrafo(gr);

    liberaGrafo(gr);

    return 0;
}

```