ROTEIRO 5 – LUCAS EDUARDO LEITE COSTA

1.1

questão-1 > C main.c > main()

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

int main(){
    Fila *F;
    int escolha, elem;
    do{
        scanf("%d", &escolha);
        switch(escolha){
            case 1:
                F = criaFila();
                break;
            case 2:
                scanf("%d", &elem);
                enfileirar(F,elem);
                break;
            case 3:
                printf("Início da fila: %d\n", verInicio(F));
                break;
            case 4:
                desenfileirar(F);
                break;
            case 5:
                imprime(F);
                break;
            case 6:
                destroiFila(F);
                break;
            case 7:
                break;
            default:
                printf("Opção inválida\n");

        }
    }while(escolha!=7);

    return 0;
}
```

questão-1 > C fila.h > ...

```c
#ifndef FILA_H
#define FILA_H
#include <stdlib.h>
#include <stdio.h>

#define MAX 100
typedef struct{
    int qtd, ini,fim;
    int dados[MAX];
}Fila;

Fila* criaFila();
void destroiFila(Fila *fi);
int tamanhoFila(Fila *fi);
int estaCheia( Fila *fi);
int estaVazia(Fila *fi);
int enfileirar(Fila *fi , int elem);
int desenfileirar(Fila *fi);
int verInicio(Fila * fi);
void imprime ( Fila * fi);

#endif
```

questão-1 > C fila.c > ...

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

Fila* criaFila(){
    Fila * fi;
    fi = (Fila*)malloc(sizeof(Fila));
    if(fi != NULL )
        fi->qtd = fi->ini = fi->fim = 0;
    return fi;
}

void destroiFila(Fila *fi){
    if(fi != NULL)
        free(fi);
}

int tamanhoFila(Fila *fi){
    if(fi == NULL )
        return -1;
    return fi->qtd;
 }

int estaCheia( Fila *fi){
    if(fi == NULL )
        return -1;
    return (fi->qtd == MAX );
}

int estaVazia(Fila *fi){
    if(fi == NULL )
        return -1;
    return (fi->qtd == 0);
}

int enfileirar(Fila *fi , int elem){
    if(fi == NULL )
        return 0;
    if(estaCheia(fi))
        return 0;
    fi->dados[fi->fim] = elem ;
    fi->fim = (fi->fim+1) % MAX ;
    fi->qtd ++;
    return 1;
}
```

questão-1 > C fila.c > ...

```c
int desenfileirar(Fila *fi){
    if(fi == NULL )
        return 0;
    if(estaVazia (fi))
        return 0;
    fi->ini = (fi-> ini +1) % MAX ;
    fi->qtd--;
    return 1;
}

int verInicio(Fila * fi){
    if(fi == NULL)
        return 0;
    if(estaVazia(fi))
        return 0;
    return fi->dados[fi->ini];
}

void imprime ( Fila * fi){
    if(fi == NULL)
        return;
    if(estaVazia (fi)){
        printf ("Fila Vazia!\n");
        return;
    }
    int i = fi->ini;
    printf ("Elementos: \n");
    do{
        printf ("%d ", fi -> dados [i]) ;
        i = (i + 1) % MAX;
    }while(i != fi->fim );
    printf ("\n");
}
```

```
[lucascosta@fedora questão-1]$ ./main
1
2
4
2
5
2
4
2
6
3
Início da fila: 4
4
3
Início da fila: 5
5
Elementos:
5 4 6
6
7
[lucascosta@fedora questão-1]$
```

1.2

questão-1.1 > C main.c > ...

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

int main(){
    Fila *F;
    int escolha, elem;
    do{
        scanf("%d", &escolha);
        switch(escolha){
            case 1:
                F = criaFila();
                break;
            case 2:
                scanf("%d", &elem);
                enfileirar(F,elem);
                break;
            case 3:
                printf("Início da fila: %d\n", verInicio(F));
                break;
            case 4:
                desenfileirar(F);
                break;
            case 5:
                imprime(F);
                break;
            case 6:
                destroiFila(F);
                break;
            case 7:
                break;
            default:
                printf("Opção inválida\n");

        }
    }while(escolha!=7);

    return 0;
}
```

questão-1.1 > C fila.h > ...

```c
#ifndef FILA_H
#define FILA_H
#include <stdio.h>
#include <stdlib.h>


typedef struct NO{
    int info ;
    struct NO* prox ;
}NO;

typedef struct{
    int qtd;
    struct NO* ini;
    struct NO* fim;
}Fila ;

Fila* criaFila();
void destroiFila(Fila *fi);
int tamanhoFila(Fila *fi);
int estaVazia(Fila *fi);
int enfileirar(Fila *fi , int elem);
int desenfileirar(Fila *fi);
int verInicio(Fila * fi);
void imprime ( Fila * fi);

#endif
```

```
[lucascosta@fedora questão-1.1]$ ./main
1
2
5
2
4
2
6
2
8
3
Início da fila: 5
4
3
Início da fila: 4
5
Elementos: 4 6 8
6
7
[lucascosta@fedora questão-1.1]$
```

questão-1.1 > C fila.c > ...

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

Fila* criaFila(){
   Fila *fi = (Fila *)malloc(sizeof(Fila));
    if (fi != NULL)
        fi->ini = fi->fim = NULL;
    return fi;
}

void destroiFila(Fila *fi){
    NO* aux = fi->ini;
    NO* aux2;
    while(aux!=NULL){
        aux2 = aux->prox;
        free(aux);
        aux = aux2;
    }
    free(fi);
}
int tamanhoFila(Fila *fi){
    int tam = 0;
    if(fi == NULL)
        return tam;
    NO* aux = fi->ini;
    while(aux != NULL){
        tam++;
        aux = aux->prox;
    }
    return tam;
}

int estaVazia(Fila *fi){
    return (fi->ini == NULL);
}
int enfileirar(Fila *fi , int elem){
    NO* n = (NO*)malloc(sizeof(NO));
    n->info = elem;
    n->prox = NULL;
    if(!estaVazia(fi))
        fi->fim->prox = n;
    else{
        fi->ini = n;
    }
    fi->fim = n;

}
```

questão-1.1 > C fila.c > ...

```c
}
int desenfileirar(Fila *fi){
    NO* aux;
    if(fi == NULL)
        return 0;
    if(estaVazia(fi)){
        printf("Fila vazia!\n");
        return 0;
    }
    aux = fi->ini;
    fi->ini= aux->prox;
    if(fi->ini == NULL)
        fi->fim = NULL;
    free(aux);
    return 1;
}
int verInicio(Fila * fi){
    if(fi == NULL)
        return 0;

    if(estaVazia(fi))
        return 0;
    int elem;
    elem = fi->ini->info;
    return elem;

}
void imprime(Fila * fi){
    if(fi == NULL)
        return;
    if(estaVazia(fi)){
        printf("Fila vazia!\n");
        return;
    }
    printf("Elementos: ");
    NO* aux = fi->ini;
    while(aux!=NULL){
        printf("%d ", aux->info);
        aux = aux->prox;
    }
    printf("\n");

}
```

2.1

questão-2 > C main.c > ...

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

int main(){
    Pilha* pi;
    int escolha, elem;
    do{
        scanf("%d", &escolha);
        switch(escolha){
            case 1:
                pi = criaPilha();
                break;
            case 2:
                scanf("%d", &elem);
                empilhar(pi,elem);
                break;
            case 3:
                printf("TOPO DA PILHA: %d\n", verTopo(pi));
                break;
            case 4:
                desempilhar(pi);
                break;
            case 5:
                imprime(pi);
                break;
            case 6:
                destroiPilha(pi);
                break;
            case 7:
                break;
            default:
                printf("Opção inválida\n");

        }
    }while(escolha!=7);

    return 0;
}
```

questão-2 > C pilha.h > ...

```c
#ifndef PILHA_H
#define PILHA_H
#include <stdio.h>
#include <stdlib.h>
#define MAX 100

typedef struct {
    int topo;
    int dados [MAX];
}Pilha;

Pilha* criaPilha();
void destroiPilha(Pilha* pi);
int tamanhoPilha(Pilha* pi);
int estaCheia(Pilha* pi);
int estaVazia(Pilha* pi);
int empilhar(Pilha* pi , int elem);
int desempilhar(Pilha* pi);
int verTopo(Pilha* pi);
void imprime(Pilha* pi);

#endif
```

questão-2 > C pilha.c > ...

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

Pilha* criaPilha(){
    Pilha * pi;
    pi = (Pilha*) malloc(sizeof(Pilha ));
    if(pi != NULL)
        pi->topo = 0;
    return pi;
}

void destroiPilha(Pilha* pi){
    if(pi != NULL )
        free(pi);
}
int tamanhoPilha(Pilha* pi){
    if(pi==NULL)
        return -1;
    return pi->topo;
}
int estaCheia(Pilha* pi){
    if(pi==NULL)
        return -1;
    return(pi->topo==MAX);
}

int estaVazia(Pilha* pi){
    if(pi==NULL)
        return -1;
    return(pi->topo==0);
}
int empilhar(Pilha* pi , int elem){
    if(pi==NULL)
        return 0;
    if(estaCheia(pi))
        return 0;
    pi->dados[pi->topo] = elem;
    pi->topo++;
    return 1;
}
```

questão-2 > C pilha.c > ...

```c
int desempilhar(Pilha* pi){
    if(pi==NULL)
        return 0;
    if(estaVazia(pi))
        return 0;
    pi->topo--;
    return 1;
}
int verTopo(Pilha* pi){
    if(pi==NULL)
        return 0;
    if(estaVazia(pi))
        return 0;
    return pi->dados[pi->topo-1];
}
void imprime(Pilha* pi){
    if(pi == NULL )
        return;
    if( estaVazia(pi)){
        printf ("Pilha Vazia!\n");
        return;
    }
    printf("Elementos: ");
    int i;
    for(i=pi ->topo -1; i >=0; i--){
        printf ("%d ", pi->dados [i]) ;
    }
    printf ("\n");
}
```

```
[lucascosta@fedora questão-2]$ ./main
1
2
3
2
5
2
4
2
6
3
TOPO DA PILHA: 6
4
5
Elementos: 4 5 3
6
7
[lucascosta@fedora questão-2]$
```

2.2

questão-2.1 > C main.c > ...

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

int main(){
    Pilha* pi;
    int escolha, elem;
    do{
        scanf("%d", &escolha);
        switch(escolha){
            case 1:
                pi = criaPilha();
                break;
            case 2:
                scanf("%d", &elem);
                empilhar(pi,elem);
                break;
            case 3:
                printf("TOPO DA PILHA: %d\n", verTopo(pi));
                break;
            case 4:
                desempilhar(pi);
                break;
            case 5:
                imprime(pi);
                break;
            case 6:
                destroiPilha(pi);
                break;
            case 7:
                break;
            default:
                printf("Opção inválida\n");

        }
    }while(escolha!=7);

    return 0;
}
```

questão-2.1 > C pilha.h > ...

```c
#ifndef PILHA_H
#define PILHA_H
#include <stdio.h>
#include <stdlib.h>


typedef struct NO{
    int info;
    struct NO* prox ;
}NO;

typedef struct {
    int qtd;
    struct NO* topo ;
}Pilha;

Pilha* criaPilha();
void destroiPilha(Pilha* pi);
int tamanhoPilha(Pilha* pi);
int estaVazia(Pilha* pi);
int empilhar(Pilha* pi , int elem);
int desempilhar(Pilha* pi);
int verTopo(Pilha* pi);
void imprime(Pilha* pi);

#endif
```

questão-2.1 > C pilha.c > ...

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

Pilha* criaPilha(){
    Pilha* pi = (Pilha*)malloc(sizeof(Pilha));
    pi->topo = NULL;
    return pi;
}
void destroiPilha(Pilha* pi){
    NO* aux;
    NO* q = pi->topo;
    while(q!=NULL){
        aux = q->prox;
        free(q);
        q = aux;
    }
    free(pi);
}
int tamanhoPilha(Pilha* pi){
    int tam = 0;
    if(pi == NULL)
        return tam;
    NO* aux = pi->topo;
    while(aux!=NULL){
        tam++;
        aux = aux->prox;
    }
    return tam;
}
int estaVazia(Pilha* pi){
    return (pi->topo==NULL);
}
int empilhar(Pilha* pi , int elem){
    NO* aux = (NO*)malloc(sizeof(NO));
    if(aux==NULL)
        return 0;
    aux->info = elem;
    aux->prox = pi->topo;
    pi->topo = aux;
    return 1;
}
```

questão-2.1 > C pilha.c > ...

```c
int desempilhar(Pilha* pi){
    NO* aux;
    if(estaVazia(pi)){
        printf("Pilha vazia!\n");
        return 0;
    }
    aux = pi->topo;
    pi->topo = aux->prox;
    free(aux);
    return 1;
}
int verTopo(Pilha* pi){
    return pi->topo->info;
}
void imprime(Pilha* pi){
    if(pi == NULL)
        return;
    if(estaVazia(pi)){
        printf("Pilha vazia!\n");
        return;
    }
    printf("Elementos: ");
    NO* aux = pi->topo;
    while(aux!=NULL){
        printf("%d ", aux->info);
        aux = aux->prox;
    }
    printf("\n");
}
```

```
[lucascosta@fedora questão-2.1]$ ./main
1
2
5
2
6
2
3
3
TOPO DA PILHA: 3
4
5
Elementos: 6 5
6
7
[lucascosta@fedora questão-2.1]$
```