INF1010 - ESTRUTURAS DE DADOS AVANCADAS - 2022.1 - 3WB

Laboratório 2 de Estruturas de Dados

Pilha com Lista Encadeada

Eric Ruas Leão - 2110694
Gusthavo Macedo - 2021030
Marina Schuler Martins - 2110075
Wladimir Ramos - 2110104

Pilha com Lista Encadeada

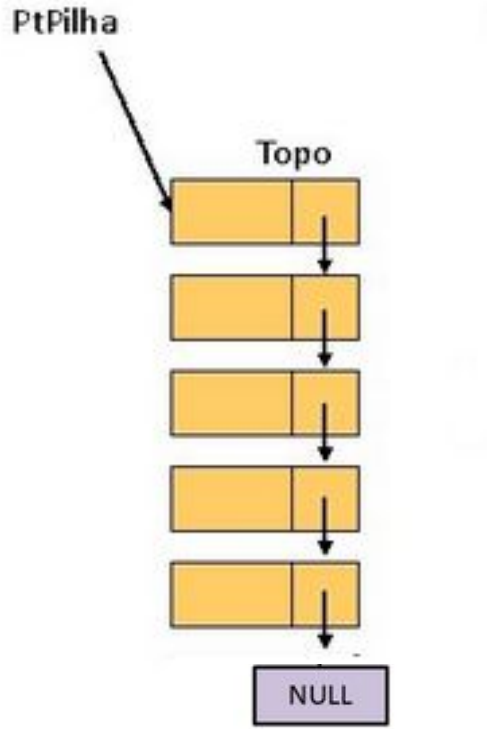
Na lista encadeada que funciona como uma pilha temos uma estrutura em que se armazena um conteúdo e um endereço de memória para a próxima posição, que vai conter o valor nulo ou o endereço para o nó que está abaixo na pilha.

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

/*estrutura da pilha*/
struct pilha {
  int info;
  struct pilha* prox;
};
typedef struct pilha Pilha;
```

Pilha com Lista Encadeada

A pilha é alterada apenas em seu topo, ou seja, novos nós são adicionados apenas na parte de cima, e para retirar um dos nós, é necessário retirar todos os que estão em cima primeiro.

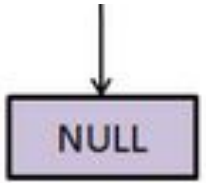


Funções Implementadas

```
Pilha *busca(Pilha * pilha,int val);
Pilha *criaPilha(void);
void liberaPilha(Pilha *pilha);
Pilha* pilha_insere (Pilha* pilha, int val);
Pilha* pilha_retira (Pilha* pilha);
int pilha_vazia (Pilha* pilha);
void imprime_pilha(Pilha *pilha);
```

Cria Pilha

```c
/*cria pilha*/
Pilha *criaPilha(void){
  return NULL;
}
```

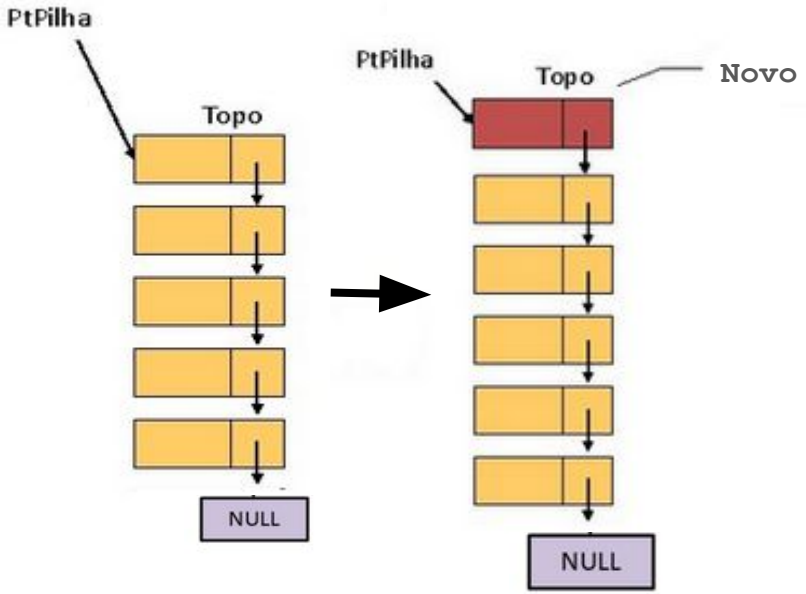


Libera Pilha

```c
/*libera pilha*/
void liberaPilha(Pilha *pilha){
  Pilha *topo = pilha;
  Pilha *proximo;
  while (topo != NULL) {
    proximo = topo->prox;
    free(topo);
    topo = proximo;
  }
}
```

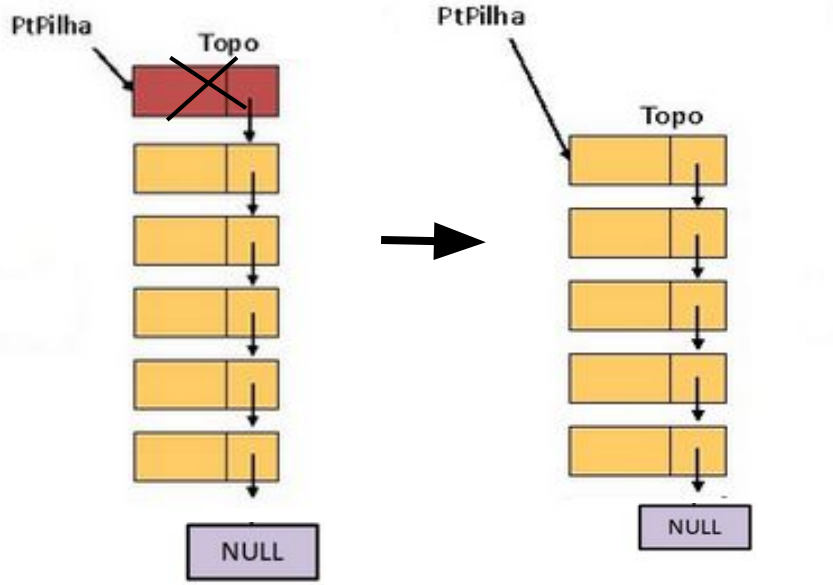
Insere na Pilha

```c
/*insere na pilha*/
Pilha* pilha_insere (Pilha* pilha, int val){
  Pilha* novo = (Pilha*) malloc(sizeof(Pilha));
  if (novo == NULL){
    printf("Stack overflow\n");
    exit(1);
  }
  novo->info = val;
  novo->prox = pilha;
  return novo;
}
```



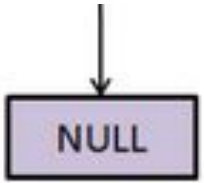
Retira da Pilha

```c
/*retira da pilha*/
Pilha* pilha_retira (Pilha* pilha){
  Pilha* aux;
  aux = pilha;
  if (aux == NULL){
    printf("Stack underflow\n");
    exit(1);
  }
  pilha = pilha->prox;
  free(aux);
  return pilha;
}
```



Verifica se a Pilha está Vazia

```c
/*verifica se pilha está vazia*/
int pilha_vazia (Pilha* pilha) {
  return (pilha == NULL);
}
```



Exibe a Pilha

```c
/*exibe a pilha*/
void imprime_pilha(Pilha *pilha){
  Pilha *p;
  for (p = pilha;p != NULL;p = p->prox)
    printf("%d,",p->info);
  printf("\n");
  return;
}
```

Busca na Pilha

```
/*busca na pilha*/
Pilha *busca(Pilha * pilha,int val){
  Pilha * aux;
  for ( aux = pilha; aux != NULL; aux = aux->prox)
    if(aux->info == val)
      return aux;
  return NULL;
}
```