1. Pilha

Definição: Em uma pilha, os ítens são colocados um sobre o outro, portanto o último ítem inserido está no topo e o primeiro ítem inserido está na base. O modelo intuitivo de uma pilha é um monte de pratos em uma prateleira.

Caso particular de lista no qual as operações de inclusão e exclusão são realizadas na mesma extremidade denominada Topo.

Estratégia LIFO (last in first out).

Quando utilizar:

1. quando qualquer um dos nós servem para utilização.
2. inverter a ordem de processamento de nós.
3. processamento de estruturas aninhadas de profundidade imprevisível, na qual deseja-se que as estruturas mais internas sejam processadas antes das mais externas.

Tipos abstratos: representação interna + operações de manipulação:

- **incluir nó na pilha**

  sobre o nó que está no topo

- **excluir nó da pilha**

  o nó que está no topo (não tem busca!!!)

- **acessar nó da pilha**

  o nó que está no topo (não tem busca!!!)

```
/* Primeiro, a definição de um tipo para o elemento */

typedef struct {
    int info;
} TNO;


/* Em seguida, um tipo para a pilha  */
typedef struct  {
    TNO vnos[MAX];
    int topo;
    int maximo;
} TPILHA;
```

```c
/*cria pilha vazia*/
  void cria_pilha_vazia (TPILHA *ppilha,int max){
      ppilha -> topo = -1;
      ppilha->max=max;

}
```

```c
//pilha está vazia?
int pilha_vazia(const TPILHA *ppilha){
     return (ppilha->topo == -1);
}
```

```c
//pilha está cheia?
int pilha_cheia(const TPILHA *ppilha){
     return (ppilha -> topo == ppilha->max-1);
}
```

```c
// Empilha ou Push
int push (TPILHA *ppilha, const TNO  *no){
     if (pilha_cheia(ppilha))
        return 0;           /*overflow*/
     (ppilha -> topo)++;
     ppilha -> vnos[ppilha -> topo] = *no;
     return 1;
}
```

```c
//Desempilha ou Pop
int pop (TPILHA *ppilha, TNO *no){
    if (pilha_vazia(ppilha))
        return 0;           /* Underflow */

    *no = ppilha -> vnos[ppilha -> topo];
    (ppilha -> topo)--;
    return 1;
}
```