

1.- ESPECIFICACIÓN:

tad Pila(telemento)

{ Pila está formada por elementos de tipo telemento }

Especificación de las operaciones:

accion iniciarPila(**sal** Pila P)
{ Inicia P como una pila vacía }

accion apilar(**e/s** Pila P, **ent** d : telemento)
{ Apila en P el elemento d }

funcion pilaVacía(Pila P) **devuelve booleano**
{ Si P está vacía devuelve VERDAD y FALSO en otro caso }

funcion cima(Pila P) **devuelve telemento**
{ Devuelve el elemento más reciente de la pila P y no modifica la pila }

accion desapilar(**e/s** Pila P)
{ Modifica la pila P eliminando el último elemento apilado }

2a.- IMPLEMENTACIÓN ESTÁTICA:

• **REPRESENTACIÓN:**

typedef telemento = tvector[MAXPILA]

constante

MAXPILA=....; { representa el tamaño máximo de la pila }

tipo

Pila = **registro**
 tvector datos
 entero indCima
freg;

• **INTERPRETACIÓN:**

indCima representa el número de elementos de la pila, los cuales se encuentran almacenados en las indCima primeras componentes. La cima se encuentra en la componente indCima-1.

• **IMPLEMENTACIÓN DE LAS OPERACIONES:**

```
accion iniciarPila( sal Pila P)
{ Inicia P como una pila vacía }
```

```
principio
```

```
    P.indCima ← 0
```

```
fin
```

```
accion apilar( e/s Pila P, ent telemento d)
{ Apila en P el elemento d }
```

```
principio
```

```
    si P.indCima < MAXPILA entonces
```

```
        P.datos[P.indCima] ← d
```

```
        P.indCima ← P.indCima + 1
```

```
    fsi
```

```
fin
```

```
funcion pilaVacía( Pila P) devuelve booleano
```

```
{ Si P está vacía devuelve VERDAD y FALSO en otro caso }
```

```
principio
```

```
    devuelve ( P.indCima = 0 )
```

```
fin
```

```
funcion cima( Pila P ) devuelve telemento
```

```
{ Devuelve el elemento más reciente de la pila P y no
  modifica la pila }
```

```
principio
```

```
    devuelve ( P.datos[P.indCima-1] )
```

```
fin
```

```
accion desapilar( e/s Pila P )
```

```
{ Modifica la pila P eliminando el último elemento
  apilado }
```

```
principio
```

```
    P.indCima ← P.indCima - 1
```

```
fin
```

Nota: esta implementación es válida para pilas de cómo mucho MAXPILA elementos

2b.- IMPLEMENTACIÓN DINÁMICA:

• **REPRESENTACIÓN:**

```
tipo
    Nodo = registro
        telemento dato
        puntero a Nodo sig
    freg;

Pila = puntero a Nodo
```

• **INTERPRETACIÓN:**

Pila es un puntero a un nodo que contiene la cima y un puntero que apunta al nodo que contiene el elemento anterior. El puntero del nodo correspondiente al primer elemento apilado apunta a NULL. Si la pila está vacía el puntero apunta a NULL.

• **IMPLEMENTACIÓN DE LAS OPERACIONES:**

```
accion iniciarPila( sal Pila P )
{ Inicia P como una pila vacía }
```

```
principio
    P ← NULL
fin
```

```
accion apilar( e/s Pila P , ent telemento d)
{ Apila en P el elemento d }
```

```
variables
    nuevo : puntero a Nodo
principio
    nuevo ← reservar(Nodo)
    si nuevo ≠ NULL entonces
        dest(nuevo).dato ← d
        dest(nuevo).sig ← P
        P ← nuevo
    fsi
fin
```

```
funcion pilaVacía( Pila P ) devuelve booleano
{ Si P está vacía devuelve VERDAD y FALSO en otro caso }
```

```
principio
    devuelve( P = NULL )
fin
```

```
funcion cima( Pila P ) devuelve telemento  
{ Devuelve el elemento más reciente de la pila P y no  
  modifica la pila }
```

```
principio  
  devuelve( dest(P).dato )  
fin
```

```
accion desapilar( e/s Pila P )  
{Modifica la pila P eliminando el último elemento  
  apilado }
```

```
variables  
  aux : puntero a Nodo
```

```
principio  
  aux ← P  
  P ← dest(aux).sig  
  liberar(aux)  
fin
```