

TDA PILA

Ing. Mario Milton López Winnipeg

TDA Pila

Pilas

- 1 Descripción del TDA Pila.
- 2 Especificación del TDA Pila.
- 3 Ejemplos de uso.
- 4 Implementaciones del TDA Pila.
 - 4.1 Implementación basada en el TDA Lista.
 - 4.2 Implementación con vectores.
 - 4.3 Implementación con apuntadores

1 Descripción del TDA Pila

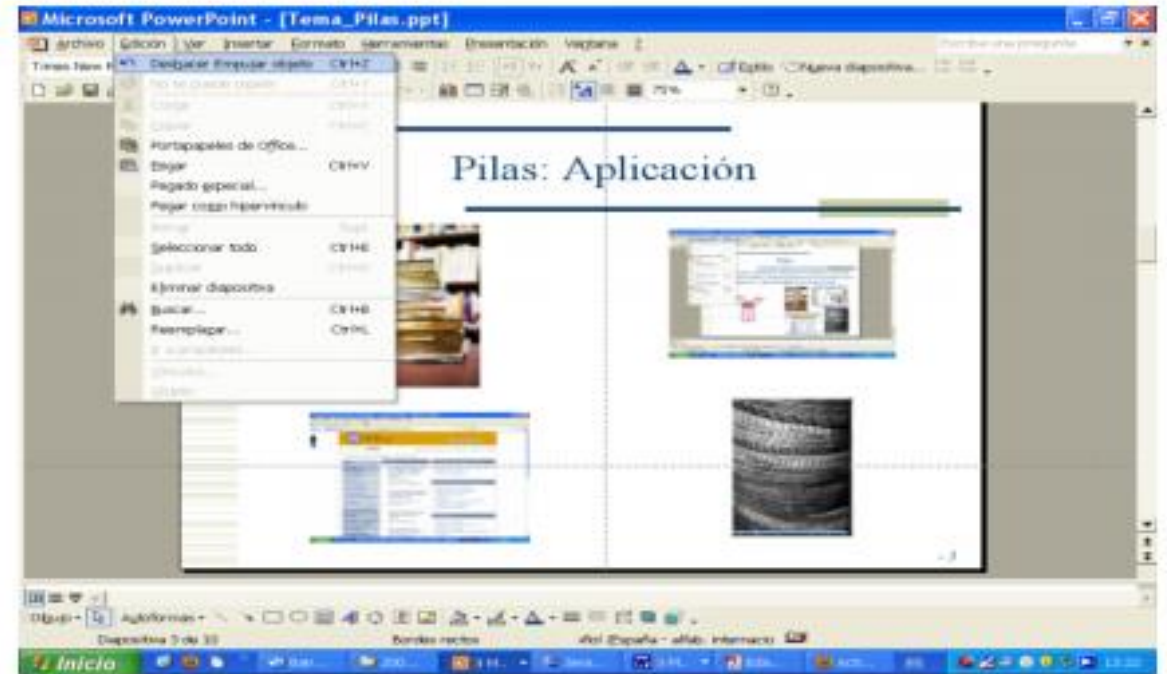
- Una pila es un caso especial de lista en la cual todas las inserciones y supresiones tienen lugar en un extremo determinado llamado *tope*.
- A las pilas se les llama también listas *LIFO* (*last-in first- out*) o listas “ultimo en entrar, primero en salir”.
- En el TDA Pila no se definen operaciones de posicionamiento en la pila. Esto es debido a que todas las operaciones de acceso se realizan en la misma posición, el tope de la pila.

METER

SACAR



Undo operaciones



Back Navigation

1 Descripción del TDA Pila

Un TDA de la familia pila incluye a menudo las cinco operaciones siguientes:

- **CREA.** Crea una pila vacía.
- **VACIA.** Devuelve un valor cierto si la pila está vacía, y falso en caso contrario.
- **CIMA.** Devuelve el elemento situado el tope de la pila, sin extraerlo.
- **METER.** Añade un elemento a la pila, quedando éste situado en el tope.
- **SACAR.** Suprime el elemento situado en el tope de la pila.

2 Especificación del TDA Pila

Especificación informal del TDA Pila

- Pila = TDA con operaciones crea, vacia, cima, meter, sacar.
- DESCRIPCIÓN:
 - Los valores del TDA Pila son pilas de elementos del tipo Elemento. Las pilas son mutables: meter y sacar añaden y eliminan elementos en la pila respectivamente.

2 Especificación del TDA Pila

OPERACIONES

- **crea(P Pila) devuelve**
 - **efecto: Devuelve la pila vacía**
- **vacía(P : Pila) devuelve (booleano)**
 - **efecto: Devuelve cierto si la pila vacía, y falso en caso contrario.**
- **cima(P : Pila) devuelve (E:Elemento)**
 - **requerimientos: La pila es no vacía.**
 - **efecto: Devuelve en E el elemento situado en el tope de la pila**
- **meter(P: Pila, E:Elemento)**
 - **Modifica : Pila**
 - **efecto: Añade el elemento E a la pila, quedando éste situado en el tope.**
- **sacar(P: Pila ,E :Elemento)**
 - **Requerimientos: La pila es no vacía.**
 - **Modifica : Pila**
 - **Efecto: Suprime el elemento situado en el tope de la pila y lo retorna**

2 Especificación del TDA Pila

Especificación Formal

□ *Tipo: Pila* (Elemento)

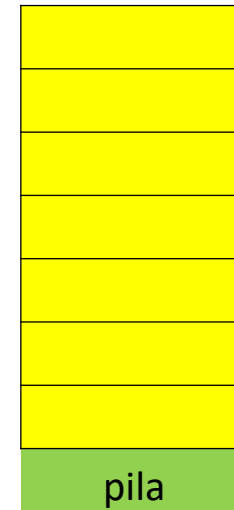
□ *Sintaxis:*

- `crea` → `Pila`
- `vacía(Pila)` → `booleano`
- `cima(Pila)` → `Elemento`
- `meter(Pila, Elemento)` → `Pila`
- `Sacar(Pila, ES Elemento)` → `Pila`

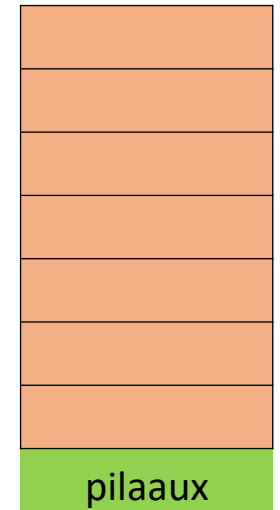
3 Ejemplos de Uso

```
publico void imprimir(ES Pila pila)
inicio

    Pila pilaAux
    mientras no (pila.vacia())
    inicio
        pila.sacar(e)
        pilaAux.meter(e)
        mostrar( e )
    fin
    mientras !(pilaaux.vacia())
    inicio
        pilaaux.sacar(e)
        pila.meter(e)
    fin
fin
```



E=



4 Implementaciones del TDA Pila

En esta sección mostraremos tres implementaciones alternativas para el TDA Pila:

1. Implementación basada en el TDA Lista; consiste en definir una Pila utilizando una **lista**
2. Implementación con vectores; utiliza un **array** para almacenar los elementos de la Pila
3. Implementación con **apuntadores** con representación con simple enlace

4.1 Implementación basada en el TDA Lista

Clase Pila

Atributos

L : Lista

Metodos

Crear()

Vacia()

Meter(E : Elemento)

Sacar(ES E: Elemento)

cima()

Fin

Constructor Pila.Crear inicio

// Crear Objeto L

fin

Pila.meter(E: Elemento)

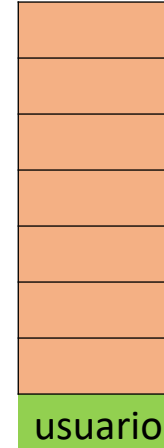
Inicio

l.inserta(l.primer(),E)

fin



implementación



4.1 Implementación basada en el TDA Lista

Clase Pila

Atributos

L : Lista

Metodos

Crear()

Vacia()

Meter(E : Elemento)

Sacar(ES E: Elemento)

cima()

Fin

Sacar (es E : Elemento)

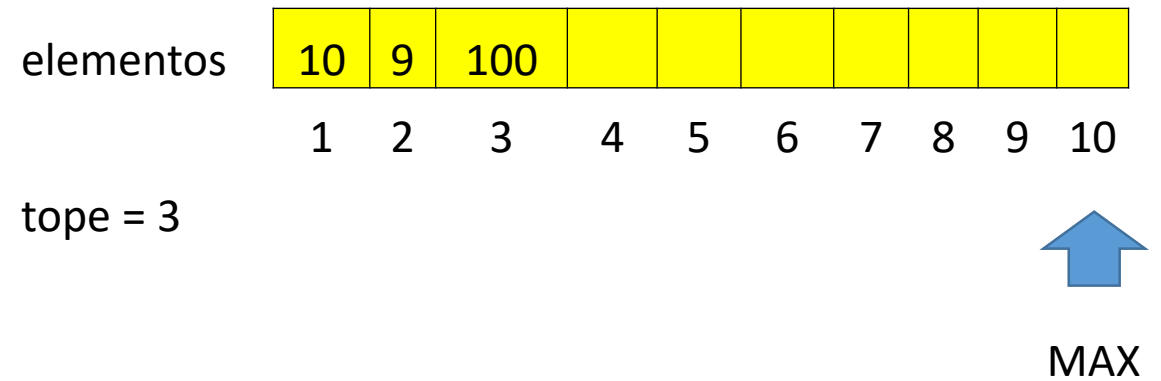
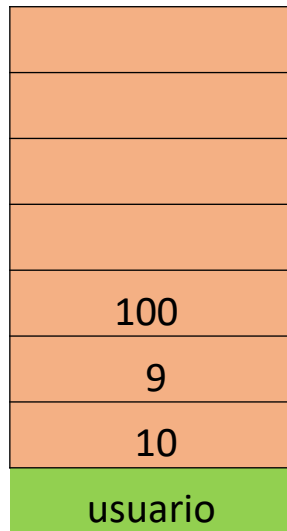
inicio

l.recupera(l.primer(),E)

l.suprime(l.primer())

fin

4.1 Implementación con vectores



4.1 Implementación con vectores

Definición básica de la clase Lista con representación contigua:

Constante max = 100

Tipo de Datos

Direccion de tipo Entero

Clase Pila

Atributos

elementos[Max] vector de tipo TipoElemento

Tope de tipo Direccion

Metodos

... .

... .

Fin

4.1 Implementación con vectores

Pila.Crear

inicio
tope = 0

Fin

Pila.vacia()

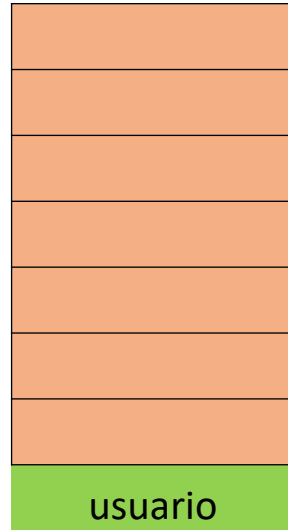
inicio
retornar (tope = 0)
Fin

Pila.meter (E : Elemento)

inicio
si tope < MAX entonces
tope = tope + 1
elementos[tope] = E
fin

Pila.Sacar (ES E: Elemento)

inicio
si no vacia() entonces
e = elementos[tope]
tope = tope - 1
caso contrario // Error
Fin



elementos

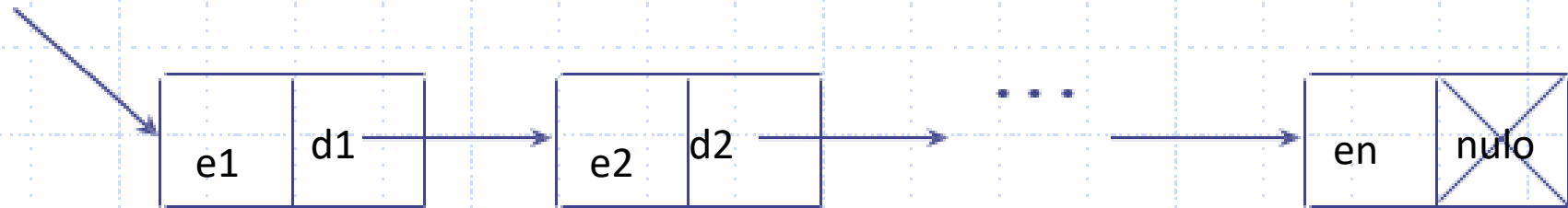


1 2 3 4 5 6 7 8 9 10

tope = 0

4.2 Implementación con apuntadores/ simulación de Memoria

Tope= DireccionMemoria



Elemento , Siguiente

4.2 Implementación con apuntadores/ simulación de Memoria

Definición básica de la clase Pila con representación enlazada con simple enlace:

Tipo de dato

Nodo

 elemento TipoElemento

 sig Puntero a Nodo

// fin definicion

Direccion Puntero a espacio de memoria de tipo Nodo

Clase Pila

 Atributos

 Tope Puntero de tipo Direccion

 Metodos

 • •

4.2 Implementación con apuntadores/ simulación de Memoria

Constructor crear

```
Inicio
  tope=-1
fin
```

publico Pila.meter(E : Elemento)

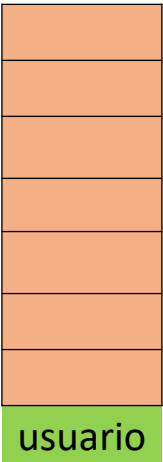
```
inicio
  aux = // Pedir Espacio de memoria para Nodo
  si aux <> nulo entoces
    ponerdato(aux,'->elemento',E)
    ponerdato(aux,'->sig',Tope)
    Tope = Aux
  caso contrario // Error
```

fin

publico Elemento Pila.cima()

```
inicio
  Si(vacia()) Entoces // Error
  caso contrario
    return obtenerdato(Tope,'->elemento')
```

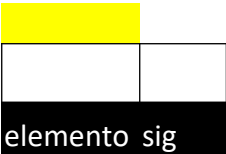
fin



Tope=-1

DIR	DATO	ID	LINK
1			2
2			3
3			4
4			5
5			6
6			7
7			8
8			9
9			10
10			11
11			12
12			13
13			14
14			15
15			-1

LIBRE=1



Aux=

4.2 Comparación de las implementaciones

- La elección de una implementación u otra dependerá de los requerimientos de la aplicación en la que se use.
- Es posible obtener representaciones para pilas que permiten operaciones eficientes que se realizan en tiempos de ejecución constantes.

LABORATORIO

- 1) CREAR MEMORIA
- 2) PEDIR ESPACIO
- 3) CREAR PILA
- 4) PONER PILA
- 5) MOSTRAR MEMORIA
- 6) CREAR CONJUNTO
- 7) PONER ELEMENTO
- 8) MOSTRAR CONJUNTO
- 9) SALIR

PUNTEROS / LISTA / VECTORES

- 1) CREAR PILA
- 2) PONER PILA
- 3) MOSTRAR PILA
- 4) SALIR