

## INTERFAZ PUBLICA DEL TAD PILA

Las operaciones apilar y desapilar constituyen las operaciones mínimas necesarias para manipular una pila.

- new: crea una pila vacía en memoria
- apilar(elem): agrega a elem en el tope de la pila
- desapilar(): saca y retorna el elemento del tope de la pila
- Otras operaciones:
- tope(): devuelve el elemento en el tope de la pila sin sacarlo de la pila.
- estaVacía(): devuelve true si la pila no tiene elementos

Ejemplo: suponer que se reciben materiales (bolsas de cal, de cemento y de pastina) en una obra en construcción y se debe simular la descarga de los mismos desde el camión (se apilan los materiales en la vereda) . Informar la cantidad total de bolsas de cemento descargadas.

```
Pila p,paux;  
string material;  
  
p=new Pila();  
  
Console.WriteLine("ingresa material?s/n");  
string res=Console.ReadLine();  
while (res == "s")  
{  
    Console.WriteLine("ingrese nombre del material: ");  
    material=Console.ReadLine();  
  
    p.apilar(material);  
    Console.WriteLine("ingresa otro material?s/n");  
  
    res=Console.ReadLine();  
}  
  
int cemento=0;  
paux=new Pila();  
  
/*imprimo la pila para ver como quedo*/  
Console.WriteLine("*****");  
while (!p.vacia())  
{
```

```

        material=p.desapilar(); /*retorna y elimina el elemento del tope*/
        Console.WriteLine(material);
        if ( material=="cemento")
            cemento++;
        paux.apilar(material); /*apilo en otra pila auxiliar */

    }
    Console.WriteLine("*****");
    Console.WriteLine("El total de bolsas de cemento apiladas es {0} ", cemento);
    Console.ReadKey();

    /*para volver a tener la pila original con los datos en el mismo orden, paso de paux a
p*/
    while (! paux.vacía())
    {
        material=paux.desapilar();
        p.apilar(material);
    }

```

Se puede resolver el problema sin saber cómo está implementada cada operación de la pila, ni siquiera saber en qué estructura de datos se almacenan los mismos en la memoria.