

# Pilas y colas

## Pilas

Una pila (stack) es un objeto similar a una pila de platos, donde se puede agregar y sacar datos sólo por el extremo superior. En computación esta estructura es utilizada ampliamente, aunque muchas veces los usuarios ni siquiera se percaten. Están tan arraigadas como concepto en computación, que una de las pocas cosas que los procesadores (CPU) saben hacer, aparte de operaciones aritméticas, es usar pilas.

La Pila se utiliza con las siguientes funciones:

- poner(p,x): agrega el elemento x a la pila p. También es conocido como *push*
- int sacar(p): quita el elemento que está en la cima de la pila p (acá suponemos una pila de números enteros). Hay gente que se refiere a este método como *pop* y *pull*.
- vacía(p): retorna 1 si la pila p no tiene ningún elemento, 0 sino.

Las pilas son estructuras LIFO (*last in, first out*), el último que entra es el primero en salir.

Barbarismos: una pila NO ES la batería que usa el reloj del computador, como alguna vez dijo un alumno de pocas luces. Tampoco es cierto lo que una vez salió en un Condorito: la invención del auto a pila (a pila de "jetones" que lo empuja)

**Propuesto:** Implemente una pila y sus funciones usando arreglos y otra usando listas enlazadas.

## Colas

Cuando uno quiere ir a comer un sandwich a la cafetería se pone en una fila. Esta estructura es una estructura FIFO (*first in, first out*), el primero en entrar es el primero en salir. En computación se llama "colas" a las filas (en la práctica también, quien no ha escuchado "¡a la cola patudo!").

La Cola se utiliza, en forma análoga a una Pila, con las siguientes funciones:

- encolar(c,x): agrega el elemento x al final de la cola c.
- int decolar(c): saca el elemento que se encuentra al principio de la cola c (acá también estamos suponiendo que la cola es de números enteros).
- int cabeza(c): devuelve cual es el elemento que esta al principio de la cola c, sin sacarlo.
- vacia(c): devuelve 1 si la cola no tiene elementos, 0 sino.

**Propuesto:** Implemente una cola y sus funciones usando arreglos y otra usando listas enlazadas.

## Algunos ejemplos

### Ejemplo 1: invertir números

Este es un programa muy simple que escribe en orden inverso una sucesión de números leídos desde el teclado:

1. Pila p;
2. int numero;
3. printf ("ingresa números positivos, termina con -1\n");

```

4. scanf("%d",&numero);

5. while (numero!=-1)

6. {

7.     poner(p,numero);

8.     scanf("%d",&numero);

9. }

10. while (!vacía(p)) print ("%d ",sacar(p));

```

Línea(s)	Ejecución
1	se crea una pila auxiliar vacía. Suponemos que se guardan enteros en la pila
4	se leen números mientras estos sean distintos de -1
7	el número se agrega a la pila por arriba
10	mientras haya números en la pila, se sacan (por arriba) y se escriben

### Ejemplo 2: una triquiñuela

Este método recibe una Pila (suponemos que se guardan enteros en ella) y la devuelve invertida, usando una cola auxiliar.

```

1. void invertir(Pila p)

2. {

3.     Cola c;

4.     while (!vacía(p))

5.     {

6.         int n = sacar(p);

7.         encolar(c,n);

8.     }

9.     while (!vacía(c))

10.     poner(p,decolar(c));

11. }

```

Línea(s)	Ejecución
10	La función invertir recibe una Pila p, de la clase PilaString. Si bien la referencia NO puede cambiar, el

	contenido del objeto puede ser modificado dentro de la función y el cambio se verá en todo el programa.
13	se crea una cola auxiliar vacía.
16-17	se saca el elemento de la pila y se pone en la cola. Esto mismo se podría haber escrito así: <code>encolar(c,sacar(p));</code>
19-20	mientras haya elementos en la cola, se van sacando y colocando en la pila

### Ejemplo 3: otra más

Análogo al ejemplo anterior, pero usando dos pilas en vez de una cola.

```

1. void invertir(Pila p)
2. {
3.   Pila aux1, aux2;
4.   while (!vacía(p))
5.     poner(aux1,sacar(p));
6.   while (!vacía(aux1))
7.     poner(aux2,sacar(aux1));
8.   while (!vacía(aux2))
9.     poner(p,sacar(aux2));
10. }
```

Línea(s)	Ejecución
22-23	Se vacía la pila p en la pila aux1
24-25	Se vacía la pila aux1 en la pila aux2
26-27	Se vacía la pila aux2 en la pila p

### Ejemplo 4: un ejemplo “real” (Homenaje a Kurt Schwarze, coautor de estos apuntes)

En un establo se guarda un rebaño de hipopótamos. Se desea ordenar los hipopótamos según su peso, usando dos establos adicionales. Los establos son tan angostos que los hipopótamos deben estar formados en fila. Cada establo tiene un abertura que es la entrada y salida al mismo tiempo.

```

1. while (!vacía(establo1))
```

```

2. {
3. Hipo h=sacar(establo1);
4. poner(establo2,h);
5. while (!vacía(establo1))
6. {
7.   Hipo h2=sacar(establo1);
8.   h=sacar(establo2);
9.   if (peso(h) < peso(h2))
10.  {
11.    poner(establo2,h);
12.    poner(establo3,h2);
13.  }
14. else
15.  {
16.    poner(establo3,h);
17.    poner(establo2,h2);
18.  }
19. }
20. while (!vacía(establo3))
21.  poner(establo1,sacar(establo3));
22. }

```

Línea(s)	Ejecución
34-35	se pasa un hipo del establo1 al establo2 (donde estan todos ordenados)
38-48	se saca un hipo del establo 1 y se compara con el hipo mas gordo del establo 2. El mas flaco se deja en el establo 2 y el otro se tira al establo 3
36	esto se hace mientras haya hipos en el establo 1
51-52	se mueven todos los hipos desde el establo3 hacia el establo1
36	esto se repite mientras haya hipos en el establo 1

**Ejemplo 5:** La boletería del cine

Para ir a ver la película "Titanic II, la ira de Rose" hay dos filas de personas. Un acomodador es el encargado de dejar entrar a la gente. Sus instrucciones son simples: "deja entrar siempre a la persona que es mayor, si tienen la misma edad, a gente de la fila 1 tiene preferencia". El siguiente código muestra el comportamiento del acomodador. Suponemos que las colas guardan referencias a Personas:

```

1. while (!vacía(cola1) || !vacía(cola2))
2. {
3.   Persona p;
4.   if (vacía(c1))
5.     p=decolar(p);
6.   else
7.     if (vacía(c2))
8.       p=decolar(c1);
9.   else
10.    {
11.      Persona p1=cabeza(c1);
12.      Persona p2=cabeza(c2);
13.      if (p1.edad <= p2.edad) p=decolar(c1);
14.    else
15.      p=decolar(c2);
16.    }
17.   println ("Se atiende a %s de %d años\n",p.nombre,p.edad);
18. }

```

Línea(s)	Ejecución
54	mientras una de las dos colas tenga gente...
57-61	si una de las colas está vacía, se deja entrar a l a gente de la otra cola
64-65	vemos quienes están parados al inicio de cada cola
66	si es menor el de la cola 1, pasa él
67-68	si no, el otro

**NUNCA** saques elementos de una pila o una cola sin verificar antes que tengan elementos.

## Problemas propuestos

**Problema 1:** Escriba una función que utilizando una o más pilas, entregue el largo (cantidad de elementos) de una cola. La función debe dejar la cola en el estado original.

**Problema 2:** Una cola puede representarse utilizando dos pilas. Escriba las funciones encolar, decolar y vacia utilizando las funciones de Pilas si el tipo de dato Cola está definido como:

```
typedef struct c
```

```
{
```

```
Pila p1;
```

```
Pila p2;
```

```
} Cola;
```