

Breve Introducción a Pilas y Colas

Tarea 5, Programación Avanzada II

Rolando Rojo Rodríguez.

March 4, 2015

Pilas

En ciencias de la computación, una pila o *LIFO* (Last in, First Out) es un tipo abstracto de datos que sirve como una colección de elementos. Este nombre hace alusión a las pilas físicas (ej. pilas de platos), con dos operadores principales:

Pilas

En ciencias de la computación, una pila o *LIFO* (Last in, First Out) es un tipo abstracto de datos que sirve como una colección de elementos. Este nombre hace alusión a las pilas físicas (ej. pilas de platos), con dos operadores principales:

(i) *push*.- Añade un elemento al final colección.

Pilas

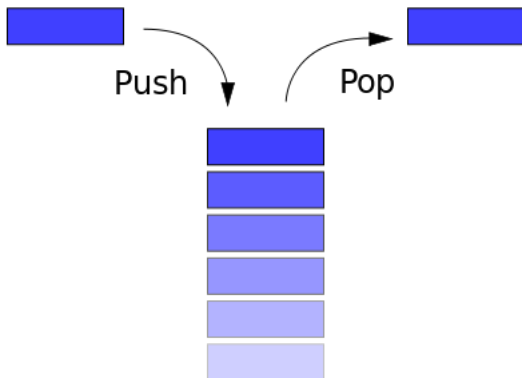
En ciencias de la computación, una pila o *LIFO* (Last in, First Out) es un tipo abstracto de datos que sirve como una colección de elementos. Este nombre hace alusión a las pilas físicas (ej. pilas de platos), con dos operadores principales:

- (i) *push*.- Añade un elemento al final colección.
- (ii) *pop*.- Remueve el último elemento añadido a la colección.

Pilas

En ciencias de la computación, una pila o *LIFO* (Last in, First Out) es un tipo abstracto de datos que sirve como una colección de elementos. Este nombre hace alusión a las pilas físicas (ej. pilas de platos), con dos operadores principales:

- (i) *push*.- Añade un elemento al final colección.
- (ii) *pop*.- Remueve el último elemento añadido a la colección.



Implementación

En lenguajes de alto nivel, una pila puede ser fácilmente implementada a través de un arreglo o una lista ligada.

Implementación

En lenguajes de alto nivel, una pila puede ser fácilmente implementada a través de un arreglo o una lista ligada.

En la librería *STL* se implementa sobre una cola de doble fin (*deque*). De tal manera que todos sus métodos son llamadas a los métodos de este último. Está implementado en base a templates por lo que es posible declarar pilas de un gran número de tipos de elementos.

Usando la librería *STL*, las pilas se declaran como:

```
stack < t > mystack;
```

donde *t* puede ser *int*, *char*, *float*, *vector*, etc.

Otros métodos y operadores Sobrecargados

Además de los métodos mencionados (*push()* y *pop()*) las pilas cuentan con otros métodos y operadores. A continuación se listan algunos de ellos:

Otros métodos y operadores Sobrecargados

Además de los métodos mencionados (*push()* y *pop()*) las pilas cuentan con otros métodos y operadores. A continuación se listan algunos de ellos:

- * *empty()*.- Regresa *True* si la pila está vacía, *False* en otro caso.

Otros métodos y operadores Sobrecargados

Además de los métodos mencionados (*push()* y *pop()*) las pilas cuentan con otros métodos y operadores. A continuación se listan algunos de ellos:

- * *empty()*.- Regresa *True* si la pila está vacía, *False* en otro caso.
- * *size()*.- Regresa el tamaño de la pila.

Otros métodos y operadores Sobrecargados

Además de los métodos mencionados (*push()* y *pop()*) las pilas cuentan con otros métodos y operadores. A continuación se listan algunos de ellos:

- * *empty()*.- Regresa *True* si la pila está vacía, *False* en otro caso.
- * *size()*.- Regresa el tamaño de la pila.
- * *back()*.- Regresa el último elemento de la pila (modo sólo lectura).

Otros métodos y operadores Sobrecargados

Además de los métodos mencionados (*push()* y *pop()*) las pilas cuentan con otros métodos y operadores. A continuación se listan algunos de ellos:

- * *empty()*.- Regresa *True* si la pila está vacía, *False* en otro caso.
- * *size()*.- Regresa el tamaño de la pila.
- * *back()*.- Regresa el último elemento de la pila (modo sólo lectura).
- * *top()*.- Regresa el primer elemento de la pila (modo sólo lectura)

Otros métodos y operadores Sobrecargados

Además de los métodos mencionados (*push()* y *pop()*) las pilas cuentan con otros métodos y operadores. A continuación se listan algunos de ellos:

- * *empty()*.- Regresa *True* si la pila está vacía, *False* en otro caso.
- * *size()*.- Regresa el tamaño de la pila.
- * *back()*.- Regresa el último elemento de la pila (modo sólo lectura).
- * *top()*.- Regresa el primer elemento de la pila (modo sólo lectura)
- * $x == y$.-Regresa *True* ssi el tamaño y los elementos de x e y son iguales.

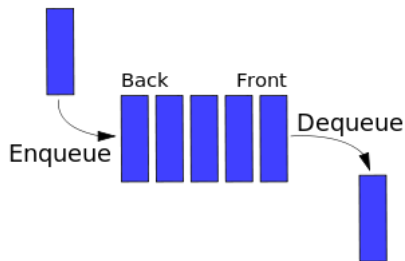
Otros métodos y operadores Sobrecargados

Además de los métodos mencionados (*push()* y *pop()*) las pilas cuentan con otros métodos y operadores. A continuación se listan algunos de ellos:

- * *empty()*.- Regresa *True* si la pila está vacía, *False* en otro caso.
- * *size()*.- Regresa el tamaño de la pila.
- * *back()*.- Regresa el último elemento de la pila (modo sólo lectura).
- * *top()*.- Regresa el primer elemento de la pila (modo sólo lectura)
- * $x == y$.-Regresa *True* ssi el tamaño y los elementos de x e y son iguales.
- * $x < y$.- Regresa *True* ssi x es lexicográficamente menor que y
- * etc...

Colas

Las colas son un tipo de contenedor abstracto de datos, específicamente diseñado para operar en contexto *FIFO* (First In, First Out). Donde los elementos son insertados en uno de los extremos del contenedor y extraídos del otro.



Implementación

De igual forma que las pilas, las colas se implementan en la librería *STL* en un contenedor *deque* en base a templates. La declaración se hace como

```
queue < t > myqueue;
```


Otros métodos y operadores Sobrecargados

Además de los métodos mencionados (*push()* y *pop()*) las pilas cuentan con otros métodos y operadores. A continuación se listan algunos de ellos:

Otros métodos y operadores Sobrecargados

Además de los métodos mencionados (*push()* y *pop()*) las pilas cuentan con otros métodos y operadores. A continuación se listan algunos de ellos:

- * *empty()*.- Regresa *True* si la cola está vacía, *False* en otro caso.
- * *size()*.- Regresa el tamaño de la cola.
- * *back()*.- Regresa el último elemento de la cola (modo sólo lectura).
- * *top()*.- Regresa el primer elemento de la cola (modo sólo lectura)
- * $x == y$.-Regresa *True* ssi el tamaño y los elementos de x e y son iguales.
- * $x < y$.- Regresa *True* ssi x es lexicográficamente menor que y
- * etc...