Representación en memoria de las pilas

Cuando se almacena una pila en la memoria de un ordenador, los elementos realmente no se mueven arriba y abajo, a medida que se meten o sacan de la pila. Simplemente es la posición del tope de la pila la que varía. Un puntero, denominado, puntero de pila, indica la posición del tope o, lo que es el mismo, el primer elemento disponible en la cima. Otro puntero se emplea para determinar la base de la pila que mantiene el valor mientras existe la pila. Figura 5.10 muestra el uso del puntero de la pila y la base de esta. Si esta se realiza la secuencia de operaciones: sacar, sacar y meter 5.9, el estado resultante de la pila aparece en la figura 5.11.Para representar una pila vacía, el puntero de la pila tiene el mismo valor que la base de la pila.

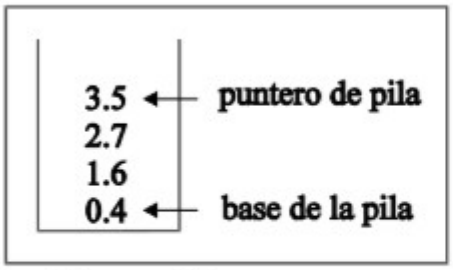


Fig. 5.10 *Pila*

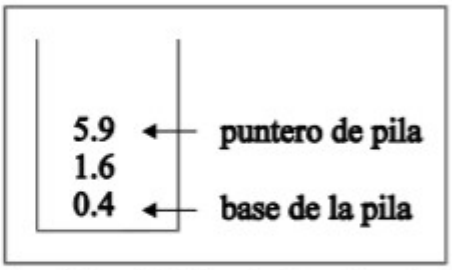


Fig. 5.11. *Estado de la pila de la Fig. 5.10 tras las operaciones: sacar, sacar, meter 5.9*

Operaciones básicas de las pilas

Insertar

En primer lugar, hay que decir que esta operación es muy comúnmente denominada push.

La inserción en una pila se realiza en su cima, considerando la cima como el último elemento insertado. Esta es una de las particularidades de las pilas, mientras el resto de las estructuras de datos lineales se representan gráficamente en horizontal, las pilas se representan verticalmente. Por esta razón es por lo que se habla de cima de la pila y no de cola de la cima. Aunque en el fondo sea lo mismo, el último elemento de la estructura de datos.

Las operaciones a realizar para realizar la inserción en la pila son muy simples, hacer que el nuevo nodo apunte a la cima anterior, y definir el nuevo nodo como cima de la pila.

Vamos a ver un ejemplo de una inserción:

Al insertar sobre esta pila el elemento 0, la pila resultante sería:

Borrar

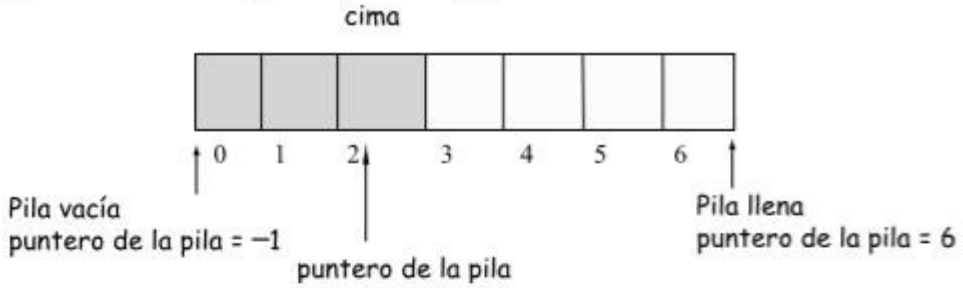
Esta operación es normalmente conocida como pop.

Cuando se elimina un elemento de la pila, el elemento que se borra es el elemento situado en la cima de la pila, el que menos tiempo lleva en la estructura.

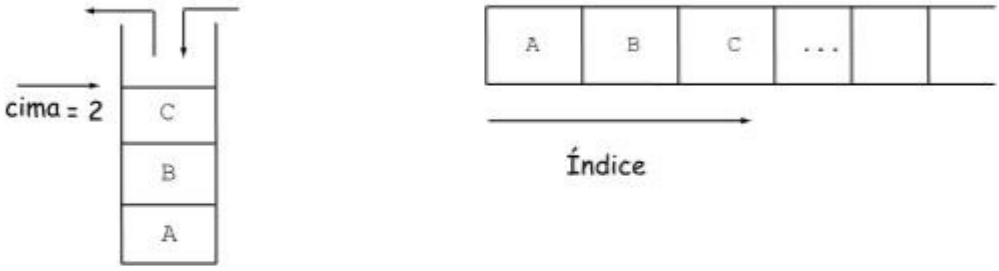
Las operaciones a realizar son muy simples, avanzar el puntero que apunta a la cima y extraer la cima anterior.

Si aplicamos la operación pop a la pila de 4 elementos representada arriba el resultado sería:

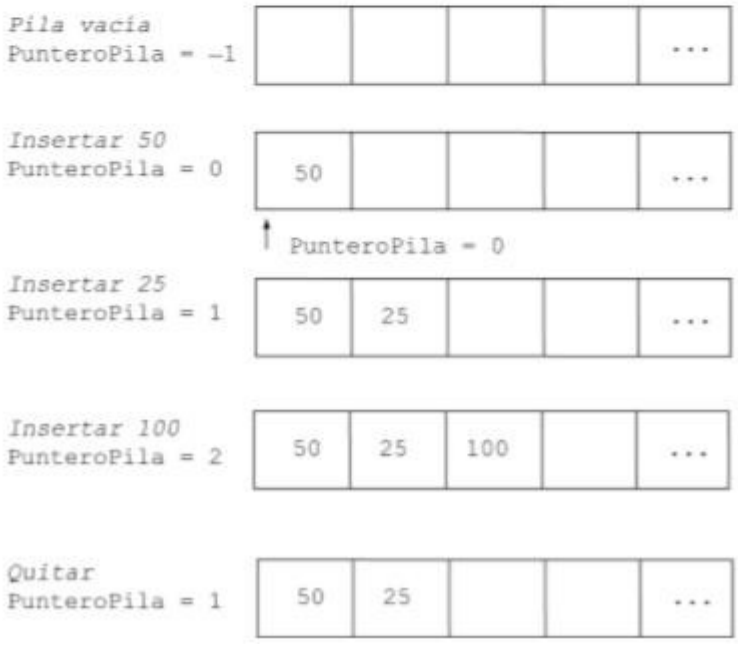
Una pila de 7 elementos se puede representar gráficamente así:



Si se almacenan los datos A, B, C, ... en la pila se puede representar gráficamente de alguna de estas formas:



A continuación se muestra la imagen de una pila según diferentes operaciones realizadas en un posible programa.



Debido a su propiedad específica último en entrar, primero en salir se conoce a las pilas como estructuras de datos LIFO (last -in, f ir st -

out) Las operaciones usuales en la pila son Insertar y Quitar. La operación Insertar (push) añade un elemento en la cima de la pila, y la operación Quitar (pop) elimina o saca un elemento de la pila. La Figura 9.2 muestra una secuencia de operaciones Insertar y Quitar. El último elemento añadido a la pila es el primero que se quita de ella.

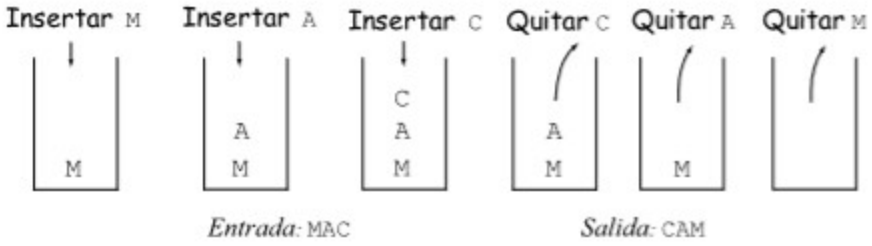


Figura 9.2 Poner y quitar elementos de la pila

La operación `Insertar` *(push)* sitúa un elemento dato en la cima de la pila, y `Quitar` *(pop)* elimina o quita el elemento de la pila.

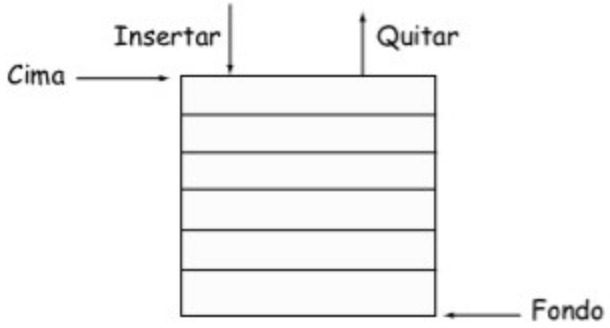


Figura 9.3 Operaciones básicas de una pila

La pila se puede implementar guardando los elementos en un array, en cuyo caso su dimensión o longitud es fija. También se puede utilizar un Vector para almacenar los elementos. Otra forma de implementación consiste en construir una lista enlazada, de modo que cada elemento de la pila forma un nodo de la lista. La lista crece o decrece según se añaden o se extraen, respectivamente, elementos de la pila; ésta es una representación dinámica, y no existe limitación en su tamaño excepto la memoria del computador. Una pila puede estar vacía (no tiene elementos) o llena (en la representación con un array —arreglo—, si se ha llegado al último elemento). Si un programa intenta sacar un elemento de una pila vacía, se producirá un error, una excepción, debido a que esa operación es imposible; esta situación se denomina desbordamiento negativo (underflow). Por el contrario, si un programa intenta poner un elemento en una pila llena, se produce un error, una excepción, de desbordamiento (overflow) o rebosamiento. Para evitar estas situaciones se diseñan métodos que comprueban si la pila está llena o vacía.

Implementación de las operaciones sobre pilas

Los métodos de la clase Pila se implementan fácilmente, teniendo en cuenta la característica principal de esta estructura: inserciones y borrados se realizan por el mismo extremo, la cima de la pila. Las operaciones insertar() y quitar() añaden y eliminan, respectivamente, un elemento de la pila; la operación cimaPila permite a un cliente recuperar los datos de la cima de la pila sin quitar realmente el elemento de la misma. La operación de insertar un elemento en la pila incrementa el puntero cima de la pila en 1 y asigna el nuevo elemento a la lista. Cualquier intento de añadir un elemento en una pila llena genera una excepción o error debido al desbordamiento de la pila.

```
public void insertar(TipoDeDato elemento)throws Exception
{
   if (pilaLlena())
   {
      throw new Exception("Desbordamiento pila");
   }
      //incrementar puntero cima y copia elemento
   cima++;
   listaPila[cima] = elemento;
}
```

La operación quitar elimina un elemento de la pila copiando, primero, el valor de la cima de la pila en una variable local, aux y, a continuación, decrementando el puntero de la pila en 1. El método quitar devuelve la variable aux, es decir el elemento eliminado por la operación. Si se intenta eliminar o borrar un elemento en una pila vacía se produce error, se lanza una excepción general.

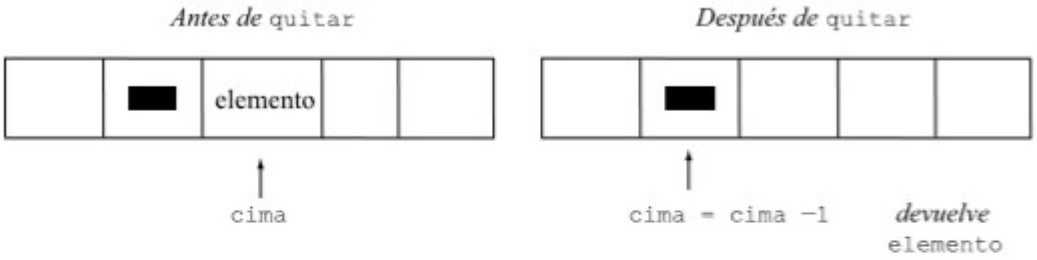


Figura 9.4 Extraer elemento `cima`

```
public TipoDeDato quitar() throws Exception
{
   TipoDeDato aux;
   if (pilaVacia())
   {
      throw new Exception ("Pila vacía, no se puede extraer.");
   }
      // guarda elemento de la cima
   aux = listaPila[cima];
      // decrementar cima y devolver elemento
   cima--;
   return aux;
}
```

La operación `cimaPila` devuelve el elemento que se encuentra en la cima de la pila. No se modifica la pila, únicamente se accede al elemento.

```
public TipoDato cimaPila() throws Exception
{
   if (pilaVacia())
   {
      throw new Exception ("Pila vacía, no hay elementos.");
   }
   return listaPila[cima];
}
```

Referencias Bibliográficas

Libro: Estructuras de datos en Java

Autores: Luis Joyanes Aguilar Ignacio Zahonero Martínez

Edificio Valrealty, 1. ª Planta

Basauri, 17 28023 Aravaca (Madrid)

Editor: José Luis García Técnico Editorial: Blanca Pecharromán

Compuesto en: Gesbiblo, S. L.

Diseño de cubierta: Gesbiblo, S. L.