Pilas

Una **pila** (stack) es una colección ordenada de elementos a los que sólo se puede acceder por un único lugar o extremo de la pila. Los elementos de la pila se añaden o quitan (borran) de la misma sólo por su parte superior (**cima**) de la pila. Éste es el caso de una pila de platos, una pila de libros, etc.

A recordar

Una pila es una estructura de datos de entradas ordenadas tales que sólo se pueden introducir y eliminar por un extremo, llamado **cima**.

Operaciones en una Pila

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.

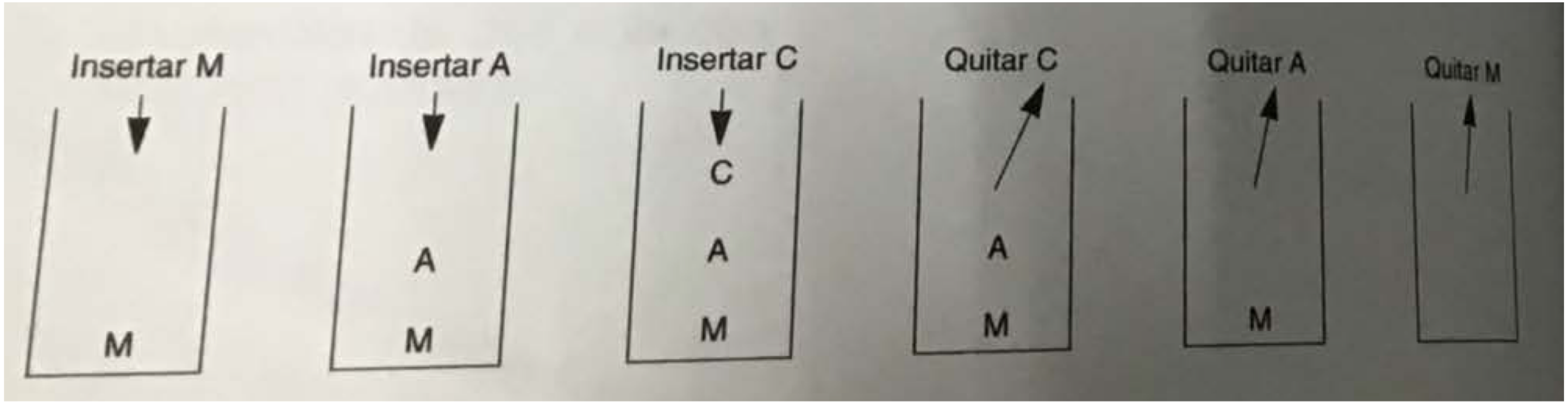


Figura: Poner y quitar elementos de la pila.

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.

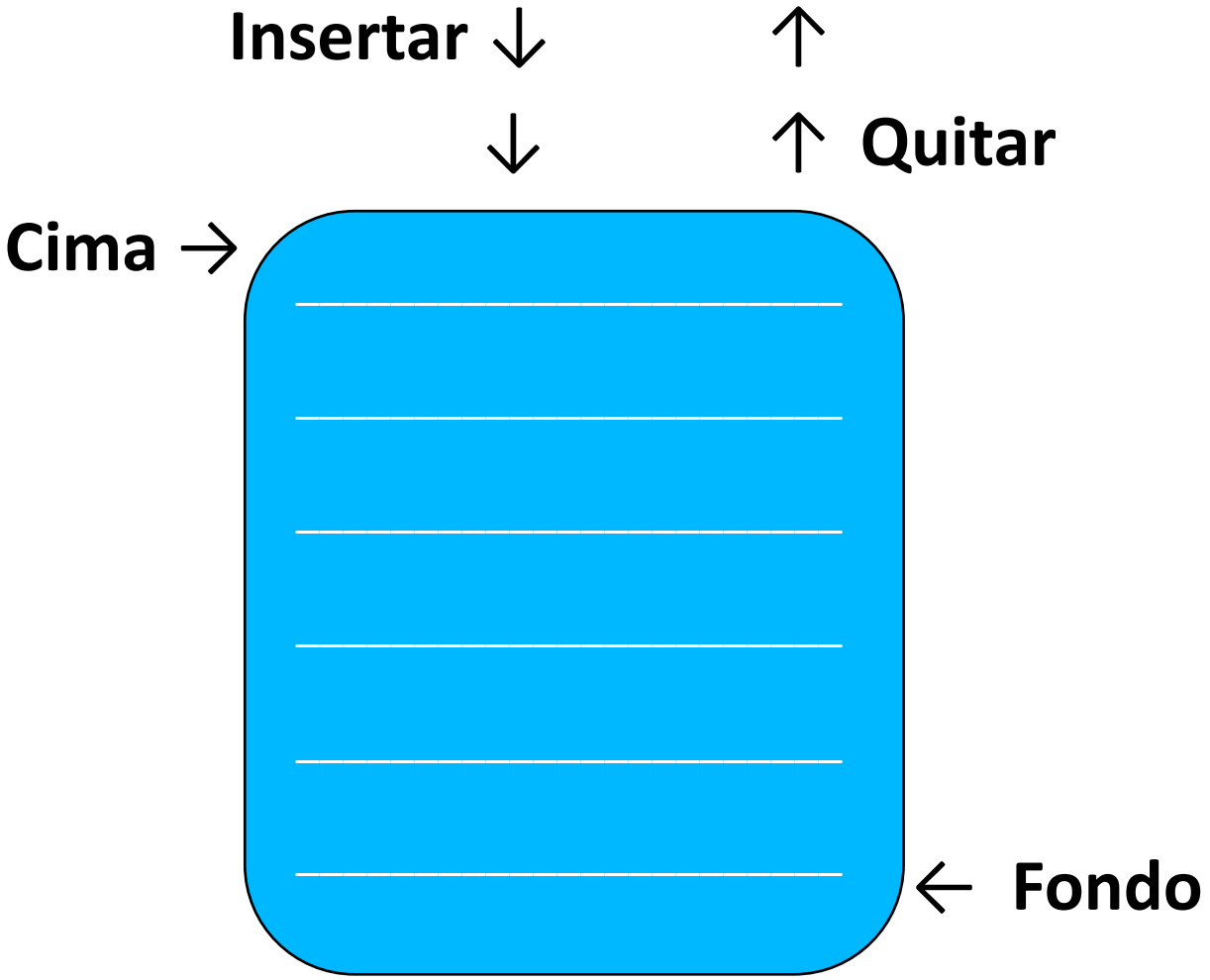


Figura: Operaciones básicas de una pila.

Especificaciones de una pila

Las operaciones que sirven para definir una pila y poder manipular su contenido son las siguientes (no todas ellas se implementan al definir una pila).

- Tipo de dato — Dato que se almacena en la pila.
- Insertar (push) — Insertar un dato en la pila.
- Quitar (pop) — Sacar (quitar) un dato de la pila.
- Pila vacia — Comprobar si la pila no tiene elementos.
- Pila llena — Comprobar si la pila está llena de elementos.
- Limpiar pila — Quitar todos sus elementos y dejar la pila vacía.
- Tamaño de la pila — Número de elementos máximo que puede contener la pila.
- Cima — Obtiene el elemento cima de la pila.

Los algoritmos de introducir <<insertar>> (push) y quitar <<sacar>> (pop) datos de la pila utilizan el índice del *array* como puntero de la pila y son:

Insertar (push)

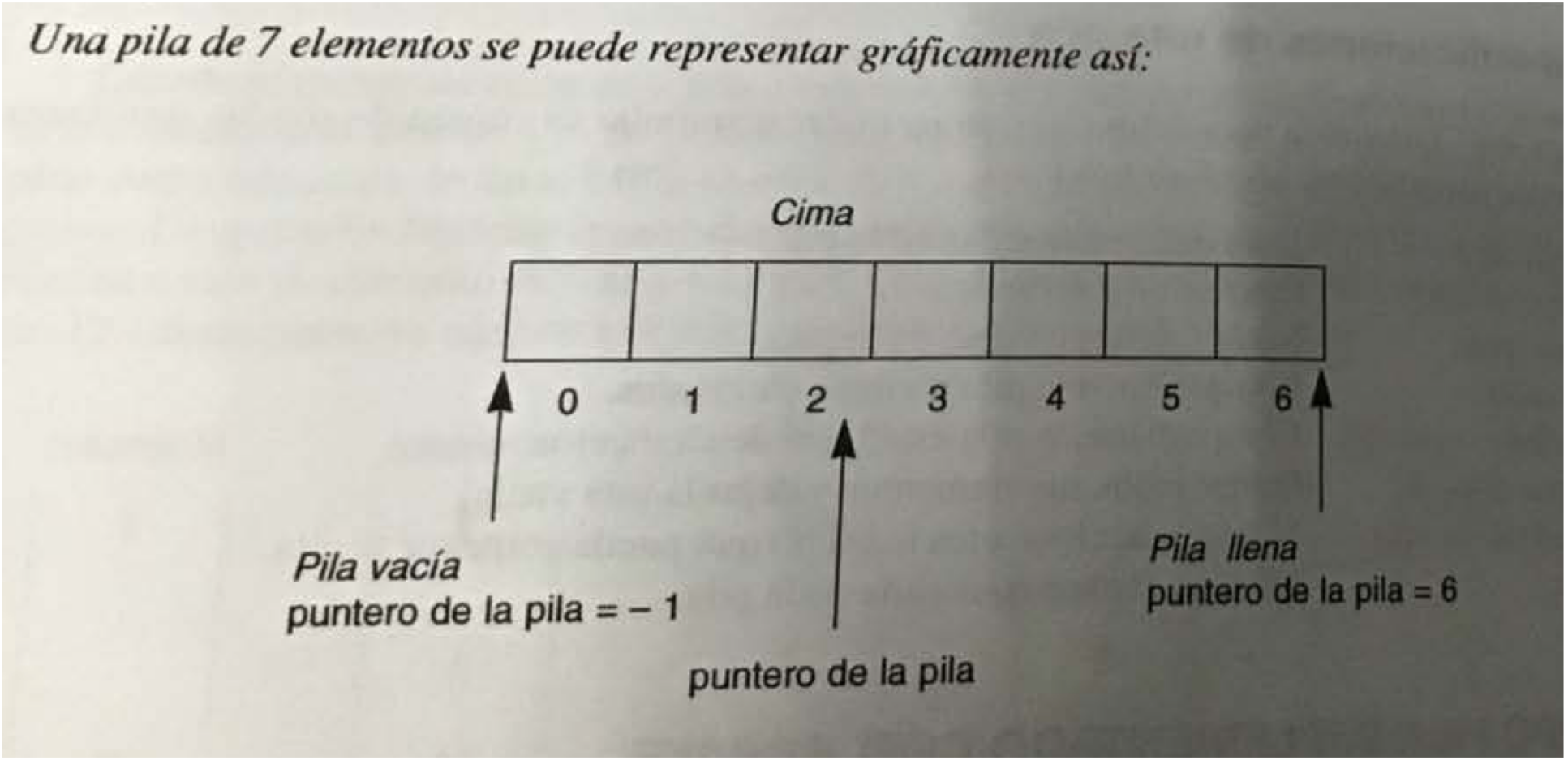
1. Verificar si la pila no está llena.
2. Incrementar en 1 el puntero de la pila.
3. Almacenar elemento en la posición del puntero de la pila.

Quitar (pop)

1. Si la pila no está vacia.
2. Leer el elemento de la posición del puntero de la pila.
3. Decrementar en 1 el puntero de la pila.

Ejemplo

Una pila de 7 elementos se puede representar gráficamente así:



pilaarray.h

El código para crear una pila require de la libreria ***pilaarray.h*** y se muestra a continuación, solo faltarían agregar las funciones.

Declaración

```
/* archivo pilaarray.h */
#include <stdio.h>
#include <stdlib.h>

#define MaxTamaPila 100

typedef struct
{
  TipoDato listapila[MaxTamaPila];
  int cima;
}Pila;

  /* Operaciones sobre la Pila */
void CrearPila(Pila* P);
void Insertar(Pila* P,const TipoDato elemento);
TipoDato Quitar(Pila* P);
void LimpiarPila(Pila* P);

  /* Operación de acceso a Pila */
TipoDato Cima(Pila P);

  /* verificación estado de la Pila */
int PilaVacia(Pila P);
int PilaLlena(Pila P);
```

Antes de incluir el archivo `pilaarray.h` debe de declararse el `TipoDato`. Así si se quiere una pila de enteros:

```
typedef int TipoDato;
#include "pilaarray.h"
```

En el caso de que la pila fuera de números complejos:

```
typedef struct
{
  float x,y;
}TipoDato;
#include "pilaarray.h"
```

Implementación de las operaciones sobre pilas

```
void CrearPila(Pila* P)
{
  P -> cima = -1;
}

void Insertar(Pila* P,const TipoDato elemento)
{

  if (P->cima == MaxTamaPila-1)
  {
    puts("Desbordamiento pila");
    exit (1);
  }
  P->cima++;
  P->listapila[P->cima] = elemento;
}

void Pop(Pila* P,TipoDato elemento)
{
  Insertar(P, elemento);
}

TipoDato Quitar(Pila *P)
{
  TipoDato aux;

  if (P->cima == -1)
  {
    puts("Se intenta sacar un elemento en pila vacía");
    exit (1);
  }
   aux = P->listapila[P->cima];
   P->cima--;
   return aux;
}
```

Implementación de las operaciones sobre pilas

```
TipoDato Cima(Pila P)
{
  TipoDato aux;
  if (PilaVacia(P))
  {
    puts("Se intenta sacar un elemento en pila vacía");
    exit (1);
  }
  aux = P.listapila[P.cima];
  return aux;
}

void LimpiarPila(Pila* P)
{
  P->cima = -1;
}

int PilaVacia(Pila P)
{
  return P.cima == -1;
}

int PilaLlena(Pila P)
{
  return P.cima == MaxTamaPila-1;
}
```

pila.c

```
#include<stdio.h>
#include<stdlib.h>
typedef int TipoDato;
#include "pilaarray.h"

void main()
{
	Pila P;
	int x,aux;

	CrearPila(&P);
	printf("Ingrese un entero en la Pila:");
	scanf("%d",&x);

	Insertar(&P,x);
	printf("\n El valor insertado en la Pila es:");
	printf("%d \n",Cima(P));
	if(!PilaVacia(P))
	printf("\n Se elimina el entero de la Pila:");
	aux=Quitar(&P);

	printf("%d \n",aux);
	LimpiarPila(&P);
}
```