

Universidad Nacional Autónoma de México

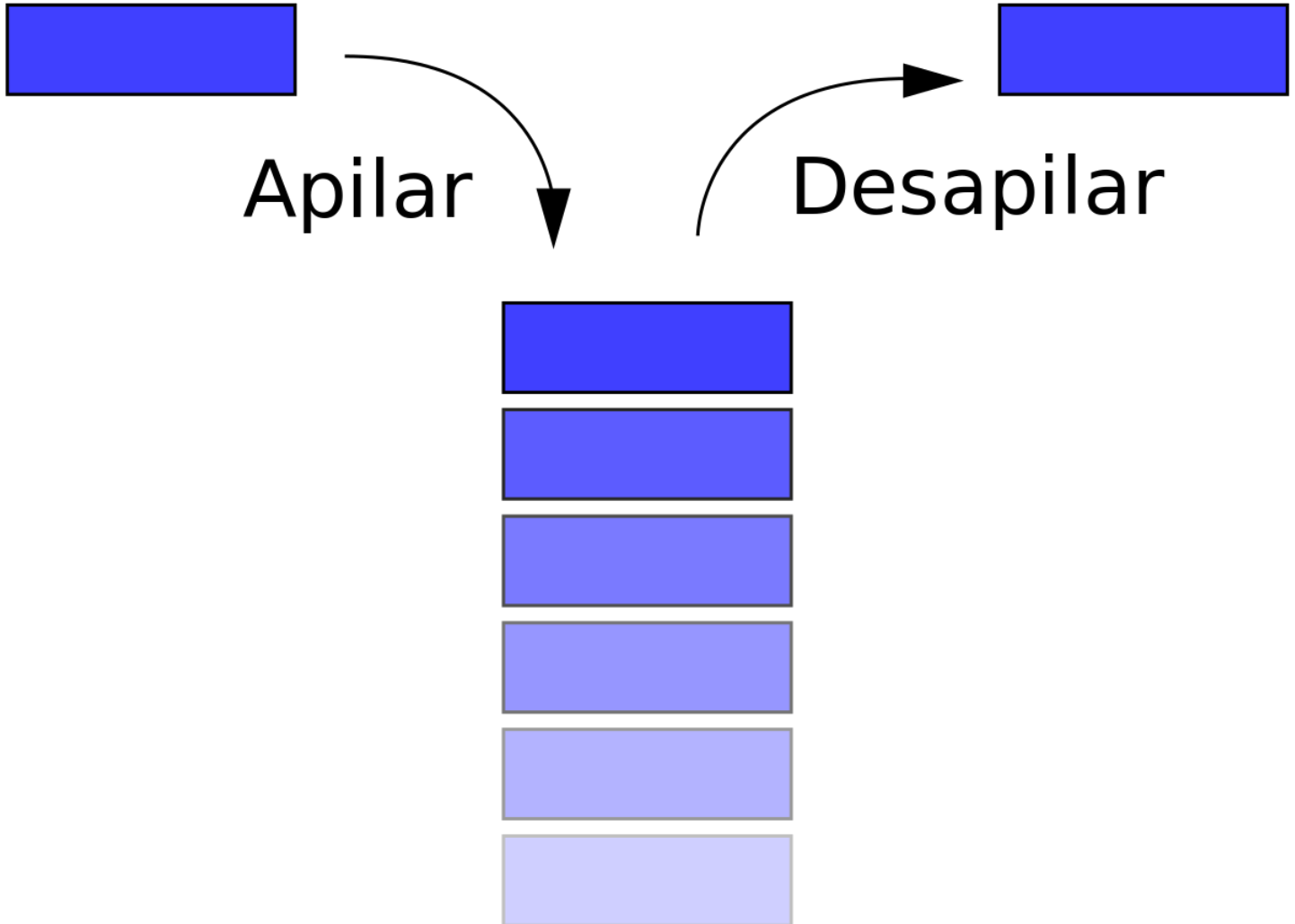
Facultad de Ingeniería

Estructura de Datos y Algoritmos I

Actividad de miércoles #6: Pila (Estructura de Datos)

Torres Oropeza Diego Alberto

09/06/2021



Definición

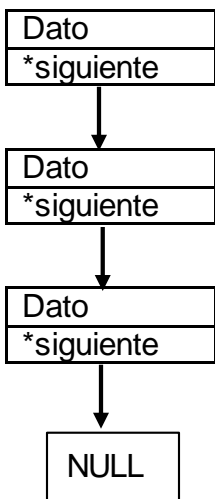
Una pila es una estructura de datos de tipo Last In. First Out, que al español se traduce como “El último en entra es el primero en salir”. En esta estructura, al ser de tipo LIFO, los datos se pueden introducir o eliminar de la pila sólo por un extremo de la pila llamado cima. Las pilas nos sirven para almacenar datos.

Operaciones

Las operaciones usuales en las pilas son:

- Insertar (push): Añade un elemento a la cima de la pila.
- Quitar (pop): Elimina o saca un elemento de la pila.

Estructura de una pila



Cada “Tipo de dato” con el “*siguiente”, es un nodo, en el que el tipo de dato puede ser cualquiera y el apuntador apunta hacia el siguiente nodo. El último nodo de forma ascendente es la cima de la pila al ser el último elemento en ella. El último nodo de manera descendente es el primer elemento de la pila y apunta a la nada, ya que, si se quita este elemento, la pila estaría vacía. La forma del nodo es esta:

```
struct Nodo{  
    int dato;  
    Nodo *siguiente;  
};
```

Insertar elementos dentro de una pila

Para insertar elementos dentro de una pila se pueden seguir 4 pasos:

Paso 1: Crear un espacio en la memoria para guardar un nodo

Al inicio de todo, la pila estará vacía:

```
pila=NULL
```

Para crear el espacio de memoria para el nuevo nodo se usa:

```
Nodo *nuevo_nodo = new Nodo();
```

Se obtiene la función para crear nuevos nodos:

```
void agregarPila(Nodo *&pila, int n){  
    Nodo *nuevo_nodo = new Nodo();  
}
```

Paso 2: Cargar el valor dentro del nodo (Dato)

Recordemos la estructura del nodo:

```
struct Nodo{  
    int dato;  
    Nodo *siguiente;  
};
```

En este paso debemos agregar un valor a dato:

```
nuevo_nodo -> dato = 10;
```

Agregamos esta línea a nuestra función:

```
void agregarPila(Nodo *&pila, int n){  
    Nodo *nuevo_nodo = new Nodo();  
    nuevo_nodo -> dato = n;  
}
```

Paso 3: Cargar el puntero pila dentro del nodo (*siguiente)

Ya rellenamos el espacio del Dato, ahora toca el del puntero, usaremos:

nuevo_nodo -> siguiente = pila;

Ahora agregamos esta línea a nuestra función:

```
void agregarPila(Nodo *&pila, int n){  
    Nodo *nuevo_nodo = new Nodo();  
    nuevo_nodo -> dato = n;  
    nuevo_nodo -> siguiente = pila;  
}
```

Paso 4: Asignar el nuevo nodo a la pila

Como nuestro nodo está señalando a nada (NULL), debemos hacer que pila señale a nuevo_nodo para que se empiecen a apilar los valores:

pila = nuevo_nodo;

La función quedaría:

```
void agregarPila(Nodo *&pila, int n){  
    Nodo *nuevo_nodo = new Nodo();  
    nuevo_nodo -> dato = n;  
    nuevo_nodo -> siguiente = pila;}  
    pila = nuevo_nodo;  
}
```

Esta función es la que nos ayudará a insertar datos dentro de la pila.

Quitar elementos de una pila

Para quitar elementos de una pila, seguiremos 4 pasos igualmente:

Paso 1: Crear una variable *aux de tipo Nodo

En este paso creamos el apuntador *aux que apuntará a pila, o sea, a la cima de la pila:

```
Nodo *aux=pila;
```

La función empezaría así:

```
void sacarPila(Nodo *&pila, int &n){  
    Nodo *aux = pila;  
}
```

Paso 2: Igualar n a aux->dato

Este paso nos permite señalar el dato del nodo de la cima, que es el que queremos quitar:

```
n = aux-> dato;
```

La función queda así:

```
void sacarPila(Nodo *&pila, int &n){  
    Nodo *aux = pila;  
    n = aux-> dato;  
}
```

Paso 3: Pasar pila al siguiente nodo

Con este paso hacemos que pila señale al siguiente nodo, porque al eliminar el nodo que ya estaba señalando, pila señalaría a nada, por eso es necesario hacer que apunte hacia el siguiente nodo:

```
pila = aux->siguiente;
```

La pila pasa a apuntar al siguiente nodo y la variable aux se queda apuntando a la cima. La función queda así:

```
void sacarPila(Nodo *&pila, int &n){  
    Nodo *aux = pila;  
    n = aux-> dato;  
    pila = aux->siguiente;  
}
```

Paso 4: Eliminar aux

Como pila ya pasó al siguiente nodo, ahora podemos eliminar el nodo de la cima (y que aux estaba apuntando) con el comando:

```
delete aux;
```

La función completa para poder quitar elementos de una pila es la siguiente:

```
void sacarPila(Nodo *&pila, int &n){  
    Nodo *aux = pila;  
    n = aux-> dato;  
    pila = aux->siguiente;  
    delete aux;  
}
```

Referencias:

- Ats, P. (2016, 14 septiembre). 95. Programación en C++ || Pilas || Concepto de Pila [Video]. YouTube. <https://www.youtube.com/watch?v=joAw2jWgZqA&feature=youtu.be>
- Ats, P. (2016b, septiembre 16). 96. Programación en C++ || Pilas || Insertar elementos a la pila [Video]. YouTube. <https://www.youtube.com/watch?v=tq3Th-zqeA4&feature=youtu.be>
- Ats, P. (2016c, septiembre 19). 97. Programación en C++ || Pilas || Quitar elementos de una pila [Video]. YouTube. <https://www.youtube.com/watch?v=YrUjHXxKz2g&feature=youtu.be>