



**UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO**



**Facultad de Ingeniería**

**Estructura de Datos y Algoritmos I**

**Profesor: M.I. Marco Antonio Martínez Quintana**

**Actividad asíncrona #02 | Operación push y pop**

**Alumna: Pineda Cruz Tania**

**No. de lista**

**Grupo: 15**

**14/06/2021**

## Algoritmo (operación PUSH)

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\push.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

espartana.c sudoku.c apuntadores.c DataAbstracto.c pilas.c almacenamiento.c push.c

1  #include <stdlib.h>
2  #include <stdio.h>
3
4  typedef struct _nodo {
5      int valor;
6      struct _nodo *siguiente;
7  } tipoNodo;
8
9  typedef tipoNodo *pNodo;
10 typedef tipoNodo *Pila;
11
12 /* Funciones con pilas: */
13 void Push(Pila *l, int v);
14 int Pop(Pila *l);
15
16 int main() {
17     Pila pila = NULL;
18
19     Push(&pila, 20);
20     Push(&pila, 10);
21     printf("%d, ", Pop(&pila));
22     Push(&pila, 40);
23     Push(&pila, 30);
24
25     printf("%d, ", Pop(&pila));
26     printf("%d, ", Pop(&pila));
27     Push(&pila, 90);
28     printf("%d, ", Pop(&pila));
29     printf("%d\n", Pop(&pila));
30
31     getchar();
32     return 0;
33 }
34
35 void Push(Pila *pila, int v) {
36     pNodo nuevo;
37
38     /* Crear un nodo nuevo */
39     nuevo = (pNodo)malloc(sizeof(tipoNodo));
40     nuevo->valor = v;
41
42     /* Añadimos la pila a continuación del nuevo nodo */
43     nuevo->siguiente = *pila;
44     /* Ahora, el comienzo de nuestra pila es en nuevo nodo */
45     *pila = nuevo;
46 }
```

## Algoritmo (operación POP)

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\push.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

espartana.c sudoku.c apuntadores.c DataAbstracto.c pilas.c almacenamiento.c push.c

1  #include <stdlib.h>
2  #include <stdio.h>
3
4  typedef struct _nodo {
5      int valor;
6      struct _nodo *siguiente;
7  } tipoNodo;
8
9  typedef tipoNodo *pNodo;
10 typedef tipoNodo *Pila;
11
12 /* Funciones con pilas: */
13 void Push(Pila *l, int v);
14 int Pop(Pila *l);
15
16 int Pop(Pila *pila) {
17     pNodo nodo; /* variable auxiliar para manipular nodo */
18     int v;      /* variable auxiliar para retorno */
19
20     /* Nodo apunta al primer elemento de la pila */
21     nodo = *pila;
22     if(!nodo) return 0; /* Si no hay nodos en la pila retornamos 0 */
23     /* Asignamos a pila toda la pila menos el primer elemento */
24     *pila = nodo->siguiente;
25     /* Guardamos el valor de retorno */
26     v = nodo->valor;
27     /* Borrar el nodo */
28     free(nodo);
29     return v;
30 }
```

## Algoritmo (operación PUSH y POP)

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\push.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

espartana.c x sudoku.c x apunadores.c x DataAbstraccio.c x pilas.c x almacenamiento.c x push.c x

1  #include <stdlib.h>
2  #include <stdio.h>
3
4  typedef struct _nodo {
5      int valor;
6      struct _nodo *siguiente;
7  } tipoNodo;
8
9  typedef tipoNodo *pNodo;
10 typedef tipoNodo *Pila;
11
12 /* Funciones con pilas: */
13 void Push(Pila *l, int v);
14 int Pop(Pila *l);
15
16 int main() {
17     Pila pila = NULL;
18
19     Push(&pila, 20);
20     Push(&pila, 10);
21     printf("%d, ", Pop(&pila));
22     Push(&pila, 40);
23     Push(&pila, 30);
24
25     printf("%d, ", Pop(&pila));
26     printf("%d, ", Pop(&pila));
27     Push(&pila, 90);
28     printf("%d, ", Pop(&pila));
29     printf("%d\n", Pop(&pila));
30
31     getchar();
32     return 0;
33 }
34
35 void Push(Pila *pila, int v) {
36     pNodo nuevo;
37
38     /* Crear un nodo nuevo */
39     nuevo = (pNodo)malloc(sizeof(tipoNodo));
40     nuevo->valor = v;
41
42     /* Añadimos la pila a continuación del nuevo nodo */
43     nuevo->siguiente = *pila;
44     /* Ahora, el comienzo de nuestra pila es en nuevo nodo */
45     *pila = nuevo;
46 }
47
48 int Pop(Pila *pila) {
49     pNodo nodo; /* variable auxiliar para manipular nodo */
50     int v;      /* variable auxiliar para retorno */
51
52     /* Nodo apunta al primer elemento de la pila */
53     nodo = *pila;
54     if(!nodo) return 0; /* Si no hay nodos en la pila retornamos 0 */
55     /* Asignamos a pila toda la pila menos el primer elemento */
56     *pila = nodo->siguiente;
57     /* Guardamos el valor de retorno */
58     v = nodo->valor;
59     /* Borrar el nodo */
60     free(nodo);
61     return v;
62 }
```

04. Símbolo del sistema - push.exe

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>gcc push.c -o push.exe
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>push.exe
10, 30, 40, 90, 20
```