



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**



FACULTAD DE INGENIERIA

Estructuras de Datos y Algoritmos I

Actividad #5 “Apuntadores en C”

Alumno: García Gallegos Luis

Grupo:15

SEMESTRE 2021-2

Fecha de entrega 26/03/2021

1. ¿Qué es un apuntador?

Los apuntadores o punteros son variables cuyo contenido es una dirección de memoria. Estos referencian direcciones de memoria de otras variables de tipo entero, flotantes, char. Un apuntador “Apunta” a la variable cuyo valor se almacena a partir de la dirección de memoria que contiene el apuntador.

Para declarar un apuntador se especifica el tipo de dato al que va apuntar, el operador*, y el nombre del apuntador, el apuntador debe ser del mismo tipo de la variable que va apuntar.

<tipo_apuntador>*<Identificador_variable>

2. ¿Dónde se aplica los apuntadores?

Tiene demasiadas ventajas al trabajar a nivel memoria, las cuales son la rapidez y la sencillez para manipular los datos. Una de sus aplicaciones es la facilidad con la que se puede recorrer un arreglo, haciendo más eficiente el acceso a los datos.

Al igual que nos permite pasar argumentos a una función y modificarlos para después regresar los argumentos modificados ya que una función solo tiene la posibilidad de regresar un dato a la función original. También sirven acceder directamente a la pantalla, al teclado, y a todos los componentes del ordenador.

3. ¿Cómo se usa en lenguaje C?

```
1 #include <stdio.h>
2 #define p printf
3 #define s scanf
4
5 int main(){
6
7     int A[10]={1,2,3,4,5,6,7,8,9,10}, i;
8     int *apuntador;
9     apuntador = &A[0];
10
11
12     //Imprime el arreglo
13     p("El arreglo esta fomrado por: ");
14     for(i=0; i<10;i++){
15         p("%3d ", *apuntador++);
16     }
17     //Imprime las direcciones de memoria
18     p("\n");
19     p("Las direcciones de memoria son: \n");
20     for(i=0; i<10;i++){
21         p("%x ", apuntador++);
22         p("\n");
23     }
24 }
```

```
El arreglo esta fomrado por:  1  2  3  4  5  6  7  8  9  10
Las direcciones de memoria son:
62fe08
62fe0c
62fe10
62fe14
62fe18
62fe1c
62fe20
62fe24
62fe28
62fe2c

-----
Process exited after 0.03783 seconds with return value 10
Presione una tecla para continuar . . . █
```

```
1  #include <stdio.h>
2  int main(){
3      int a=37,b;
4      int *p;
5      p=&a;
6      //Imprime el valor de a
7      printf("El valor de a es %d \n", *p);
8      //Multiplica el valor de a por 2
9      b=*p*2;
10     //Imprime le valor de b
11     printf("El valor de b es %d \n", b);
12
13 }
```

```
El valor de a es 37
El valor de b es 74

-----
Process exited after 0.03484 seconds with return value 21
Presione una tecla para continuar . . .
```

```

1  #include <stdio.h>
2  struct empleado
3  {
4      char nombre[35];
5      int edad;
6  };
7  int main(){
8      empleado dato;
9      empleado *pdata=&dato;
10     (*pdata).edad = 19; //Asignación del campo "edad" utilizando el operador.
11     pdata -> edad = 19; //Asignación del campo "edad" utilizando el operador "->"
12     printf("%d", pdata->edad);
13 }

```

```

19
-----
Process exited after 0.03509 seconds with return value 0
Presione una tecla para continuar . . .

```

```

1  #include <stdio.h>
2  int main(){
3      int a=37;
4      int *p,*q;
5      p=&a; //Se le asigna a p la dirección de la variable a
6      q=p; // Se le asigna a q la dirección almacenada en p (la misma de a)
7      printf("El valor de a es: %d", *q);
8  }

```

```

El valor de a es: 37
-----
Process exited after 0.03378 seconds with return value 20
Presione una tecla para continuar . . .

```

Bibliografía

- Corona Nakamura María Adriana, Ancona Valdez María de los Ángeles. (2011). Diseño de Algoritmos y su Codificación en Lenguaje C. México: Mc Graw Hill.
- Cairó Osvaldo. (2005). Metodología de la programación. México: Alfaomega.