



**Universidad Nacional
Autónoma de México**



Facultad de ingeniería

ESTRUCTURA DE DATOS Y ALGORITMOS I

Actividad 5

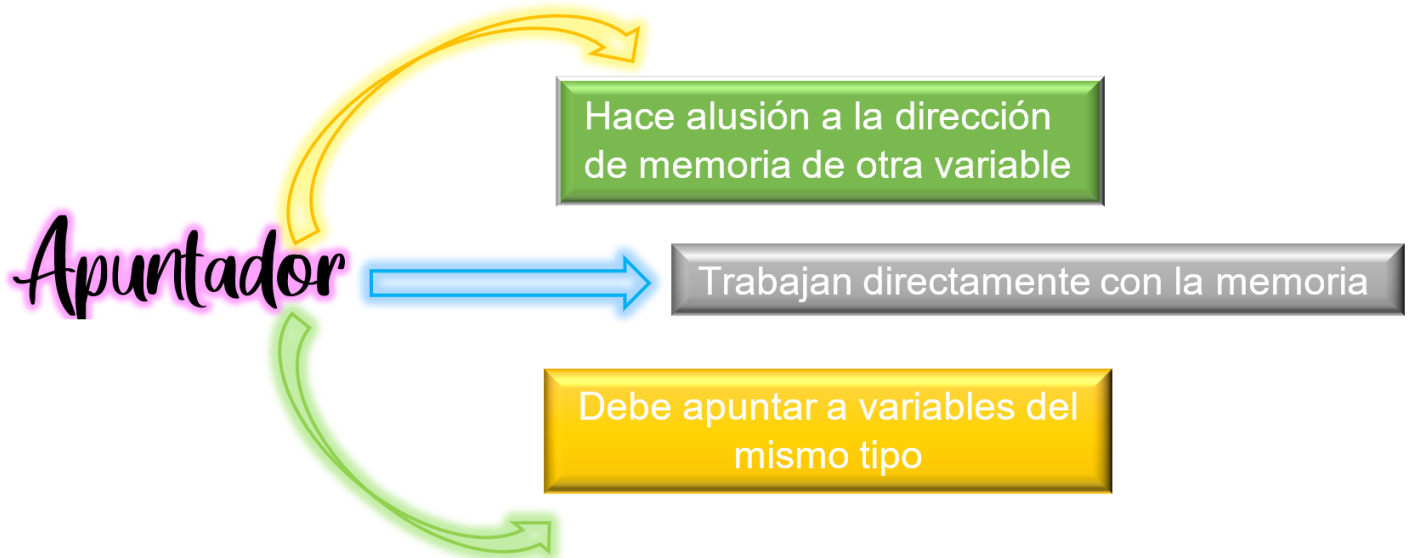
Apuntadores en C

Sánchez García Rocío

24/03/2021

- Investigar que es un apuntador

Se trata de una variable que contiene una dirección de memoria.



- ¿Dónde se aplican los apuntadores?

Se aplican principalmente en los arreglos ya que hace más sencillo el acceso a cada una de sus localidades.

- Como se usa en lenguaje C

Se debe definir el tipo de la variable y nombre separadas por un asterisco (*).

```
TipoDeDato*apuntador, variable;
```

Se debe de asignar el valor al apuntador y se accederá a la localidad de memoria de la variable a través de un ampersand (&).

```
apuntador = &variable;
```

- Ejemplo:

```

1  #include<stdio.h>
2  /*
3   El programa solicita un valor para una cadena de caracteres,
4   un número entero y uno flotante.
5   Posteriormente se imprimen los valores y direcciones en
6   memoria de cada valor
7  */
8  void main()
9  {
10     //Declaración de variables
11     char cadena[100];
12     int entero;
13     float fl;
14     //Solicitar los valores para las variables
15     printf("\n\tIngresa el valor para la cadena:");
16     scanf("%s",&cadena);
17     printf("\n\tIngresa un valor para el número entero: ",163);
18     scanf("%d",&entero);
19     printf("\n\tIngresa un valor para el número flotante: ",163);
20     scanf("%f",&fl);
21     //Declaración de variables de tipo entero
22     char*apuntadorCadena;
23     int*apuntadorEntero;
24     float*apuntadorFl;
25     //Asignar el valor a las variables
26     apuntadorCadena = cadena;//No se agrega & por que se trata de una cadena
27     apuntadorEntero = &entero;
28     apuntadorFl = &fl;
29     //Impresión de los valores a través de los apuntadores
30     printf("\n\tApuntador para su cadena: %s", apuntadorCadena);//En cadena no se agrega asterisco
31     printf("\n\tApuntador para su número entero: %d", 163,*apuntadorEntero);
32     printf("\n\tApuntador para su número flotante: %f", 163,*apuntadorFl);
33     //Impresión de las direcciones en memoria de cada una de las variables
34     printf("\n\tDirección en memoria para la cadena: %p",162,&cadena);//Con & se accede a la posición en memoria
35     printf("\n\tDirección en memoria para el número entero: %p",162,163,&entero);
36     printf("\n\tDirección en memoria para el número flotante: %p\n",162,163,&fl);
37 }

```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc apuntadoresx.c -o apuntadoresx.exe
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>apuntadoresx.exe
```

```
Ingresa el valor para la cadena:bienvenido
```

```
Ingresa un valor para el número entero: 10
```

```
Ingresa un valor para el número flotante: 9.5
```

```
Apuntador para su cadena: bienvenido
```

```
Apuntador para su número entero: 10
```

```
Apuntador para su número flotante: 9.500000
```

```
Dirección en memoria para la cadena: 0061FEB0
```

```
Dirección en memoria para el número entero: 0061FEAC
```

```
Dirección en memoria para el número flotante: 0061FEA8
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

Referencias

Manual de prácticas del Laboratorio de Estructura de Datos y Algoritmos I, Facultad de ingeniería UNAM, recuperada el 23 de marzo de 2021 en <http://lcp02.fi-b.unam.mx/>

Programación avanzada en C | Primer ejercicio con punteros y direcciones en memoria | Ejercicio #12. Kevin Korduner. 14 oct. 2017. Consultado el 24 de marzo de 2021 en: <https://www.youtube.com/watch?v=c0KPVtrN3i4>