



# Universidad Nacional Autónoma de México

## Facultad de Ingeniería



**Asignatura:**

Estructura de Datos y Algoritmos I

Actividad #5 | Apuntadores en C

**Nombre del Alumno:**

Sánchez Estrada Angel Isaac

**Maestro:**

M.I. Marco Antonio Martínez Quintana

**Grupo:**

15

**Fecha:**

26/03/2021



## **APUNTADORES EN C**

### **¿Qué es un apuntador?**

Un apuntador también llamados punteros, es una variable que contiene la dirección de una variable, es decir, hace referencia a la localidad de memoria de otra variable.

Se dice que los punteros “apuntan” a la variable cuyo valor se almacena a partir de la dirección de la memoria que contiene el apuntador.

La sintaxis para declarar un apuntador y para asignarle la dirección de memoria de otra variable es, respectivamente:

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

```
apuntador = &variable;
```

- El Operador de Dirección ( & ) regresa la dirección de una variable.
- El Operador de Indirección ( \* ), toma la dirección de una variable y regresa el dato que contiene esa dirección.

La declaración de una variable apuntador inicia con el carácter ( \* ). Cuando a una variable le antecede un ampersand, lo que se hace es acceder a la dirección de memoria de la misma (es lo que pasa cuando se lee un dato con scanf).

Los apuntadores solo pueden apuntar a direcciones de memoria del mismo tipo de dato con el que fueron declarados; para acceder al contenido de dicha dirección, a la variable apuntador se le antepone ( \* ).

### **¿Dónde se aplican?**

Los punteros se utilizan principalmente para la construcción de referencias, que a su vez son fundamentales para la construcción de casi todas las estructuras de datos, así como para pasar datos entre las diversas partes de un programa.

Debido a que los apuntadores trabajan directamente con la memoria, para acceder a ellos con rapidez para obtener un dato, se suelen utilizar para dar claridad y simplicidad a las operaciones a nivel de la memoria, por otra parte, se ocupan de igual forma para pasar parámetros por referencia. Esto es útil si el programador quiere modificaciones de una función a un parámetro.

## Como se usa en Lenguaje C (Ejemplo)

### Opción 1:

Ordena una los datos ingresados de una matriz multidimensional para verla como una unidimensional.

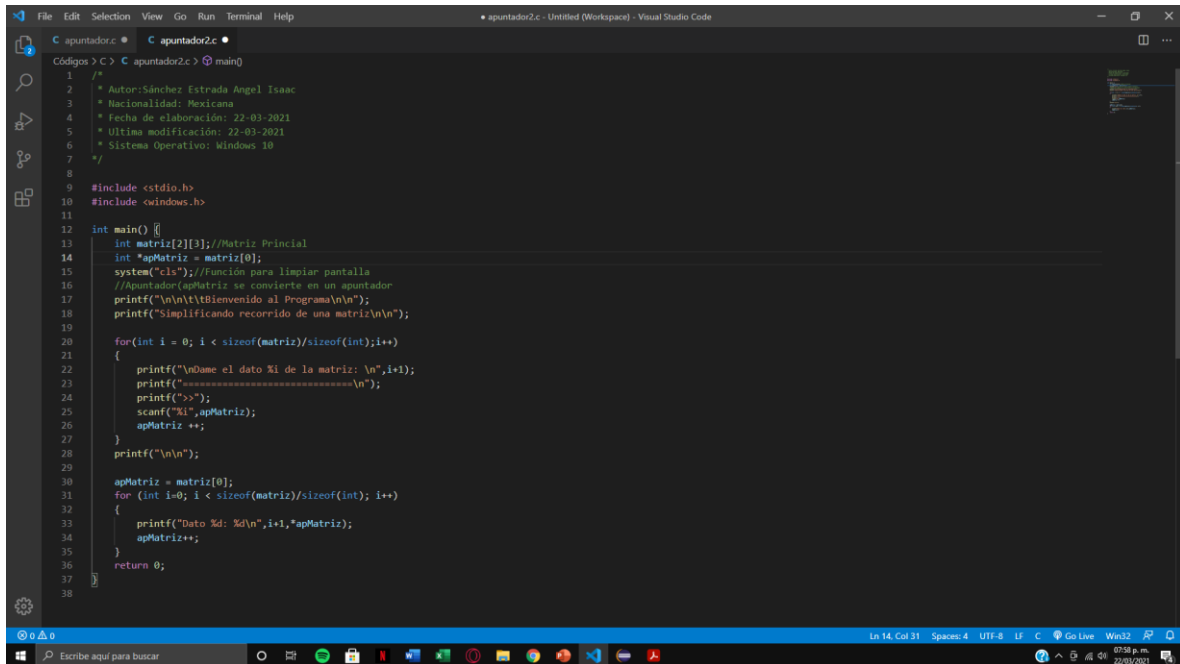
### Código Fuente

```
/*
 * Autor: Sánchez Estrada Angel Isaac
 * Nacionalidad: mexicana
 * Fecha de elaboración: 22-03-2021
 * Ultima modificación: 22-03-2021
 * Sistema Operativo: Windows 10
 */
#include <stdio.h>
#include <windows.h>

int main() {
    //Declaración de matrices
    int matriz[2][3]; //Matriz Princial
    int *apMatriz = matriz[0];
    system("cls"); //Función para limpiar pantalla
    //Apuntador(apMatriz se convierte en un apuntador
    printf("\n\n\t\tBienvenido al Programa\n\n");
    printf("Simplificando recorrido de una matriz\n\n");
    //Función para solicitar los valores que se ingresen
    for(int i = 0; i < sizeof(matriz)/sizeof(int); i++)
    {
        printf("\nDame el dato %i de la matriz: \n", i+1);
        printf("=====\n");
        printf(">> ");
        scanf("%i", apMatriz);
        apMatriz ++;
    }
    printf("\n\n");

    //Función que imprime los valores registrados
    apMatriz = matriz[0];
    for (int i=0; i < sizeof(matriz)/sizeof(int); i++)
    {
        printf("Dato %d: %d\n", i+1, *apMatriz);
        apMatriz++;
    }
    return 0;
}
```

## Código – Visual Studio Code



```
1  /*
2  * Autor: Sánchez Estrada Angel Isaac
3  * Nacionalidad: Mexicana
4  * Fecha de elaboración: 22-03-2021
5  * Última modificación: 22-03-2021
6  * Sistema Operativo: Windows 10
7  */
8
9  #include <stdio.h>
10 #include <windows.h>
11
12 int main() {
13     int matriz[2][3]; // Matriz Principal
14     int *apMatriz = matriz[0];
15     system("cls"); // Función para limpiar pantalla
16     // Apuntador apMatriz se convierte en un apuntador
17     printf("\n\n\t\tBienvenido al Programa\n\n");
18     printf("Simplificando recorrido de una matriz\n\n");
19
20     for(int i = 0; i < sizeof(matriz)/sizeof(int); i++)
21     {
22         printf("\nDame el dato %i de la matriz: \n", i+1);
23         printf("===== \n");
24         scanf("%i", apMatriz);
25         apMatriz++;
26     }
27     printf("\n\n");
28
29     apMatriz = matriz[0];
30     for (int i=0; i < sizeof(matriz)/sizeof(int); i++)
31     {
32         printf("Dato %d: %d\n", i+1, *apMatriz);
33         apMatriz++;
34     }
35     return 0;
36 }
37
38
```

## Ejecución

```

                                     Bienvenido al Programa

Simplificando recorrido de una matriz

Dame el dato 1 de la matriz:
=====
>> 5

Dame el dato 2 de la matriz:
=====
>> 8

Dame el dato 3 de la matriz:
=====
>> 4

Dame el dato 4 de la matriz:
=====
>> 3

Dame el dato 5 de la matriz:
=====
>> 1

Dame el dato 6 de la matriz:
=====
>> 7

Dato 1: 5
Dato 2: 8
Dato 3: 4
Dato 4: 3
Dato 5: 1
Dato 6: 7
```

## Opción 2

Te dice las características de la variable que defines en el código

### Código Fuente

```
#include <stdio.h>

/*
 * Autor: Sánchez Estrada Angel Isaac
 * Nacionalidad: mexicana
 * Fecha de elaboración: 22-03-2021
 * Última modificación: 22-03-2021
 * Sistema Operativo: Windows 10
 */

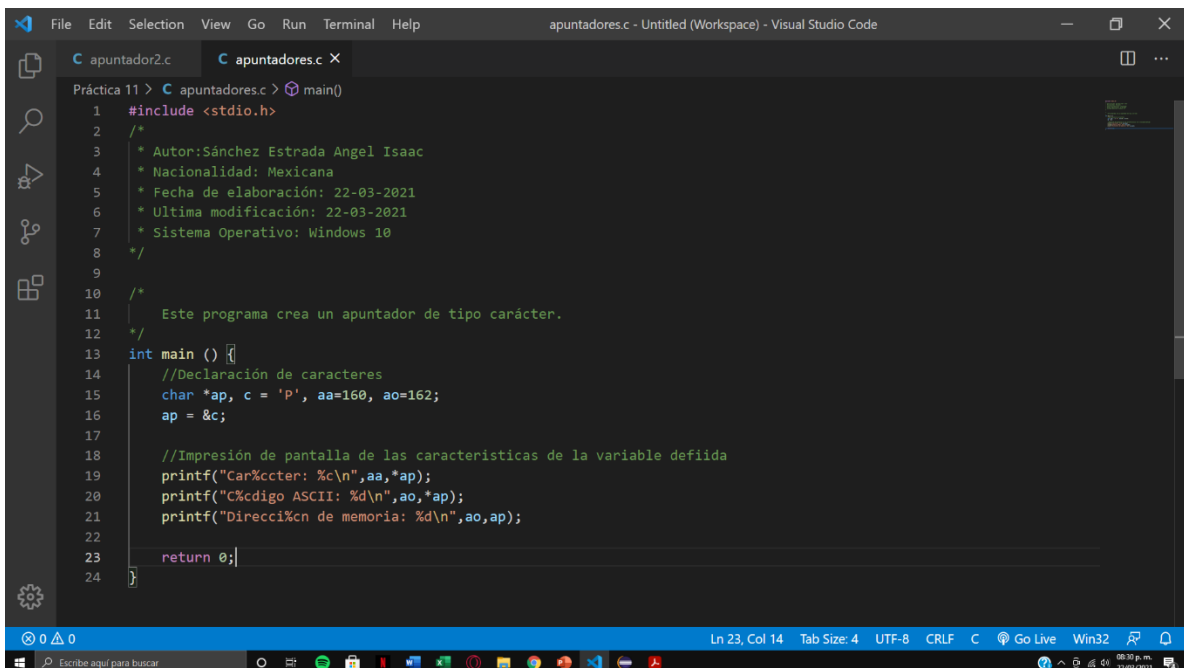
/*
 Este programa crea un apuntador de tipo carácter.
 */

int main () {
    //Declaración de caracteres
    char *ap, c = 'P', aa=160, ao=162;
    ap = &c;

    //Impresión de pantalla de las características de la variable definida
    printf("Carácter: %c\n",aa,*ap);
    printf("Código ASCII: %d\n",ao,*ap);
    printf("Dirección de memoria: %d\n",ao,ap);

    return 0;
}
```

### Código – Visual Studio Code



## Ejecución

En este caso la variable que se asignó fue P

```
Carácter: P
Código ASCII: 80
Dirección de memoria: 6422295
```

## Opción 3

Muestra el como funciona un apuntador para caracteres y como registra cada carácter por separado

## Código Fuente

```
/*
 * Autor: Sánchez Estrada Angel Isaac
 * Nacionalidad: Mexicana
 * Fecha de elaboración: 22-03-2021
 * Última modificación: 22-03-2021
 * Sistema Operativo: Windows 10
 */

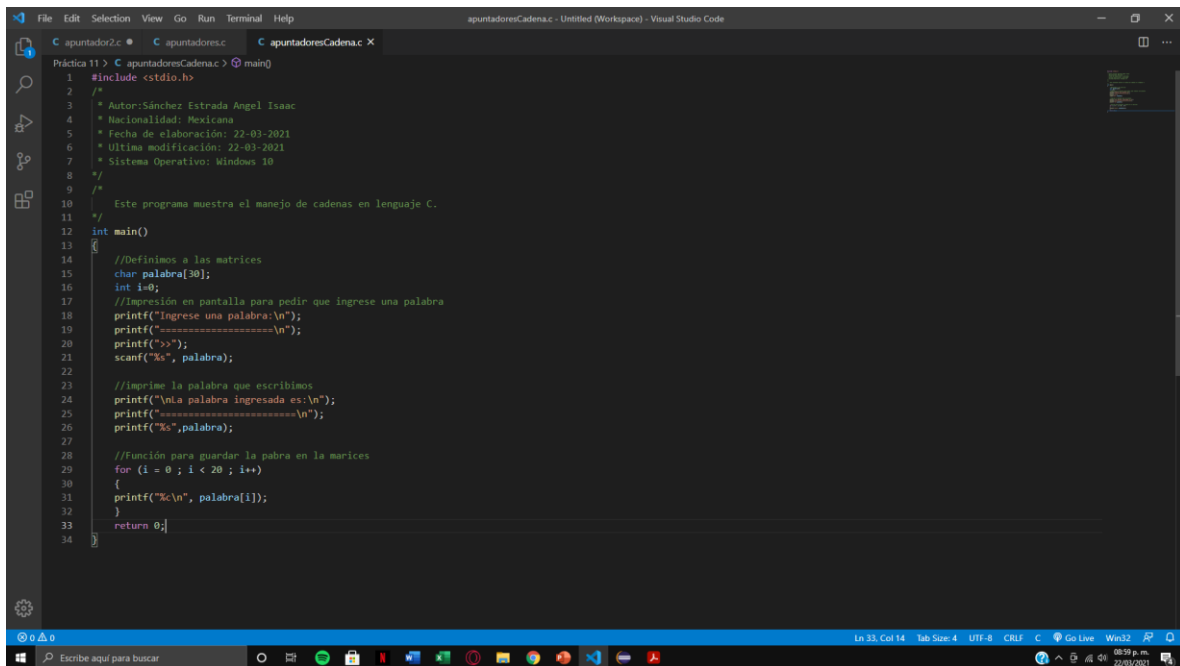
#include <stdio.h>
#include <windows.h>

int main()
{
    //Definimos a las matrices
    char palabra[30];
    int i=0;
    //Impresión en pantalla para pedir que ingrese una palabra
    printf("Ingrese una palabra:\n");
    printf("=====\n");
    printf(">>");
    scanf("%s", palabra);

    //imprime la palabra que escribimos
    printf("\nLa palabra ingresada es:\n");
    printf("=====\n");
    printf("%s", palabra);

    //Función para colocar la palabra ingresada de forma vertical
    for (i = 0 ; i < 20 ; i++)
    {
        printf("\n\n%c\n", palabra[i]);
    }
    return 0;
}
```

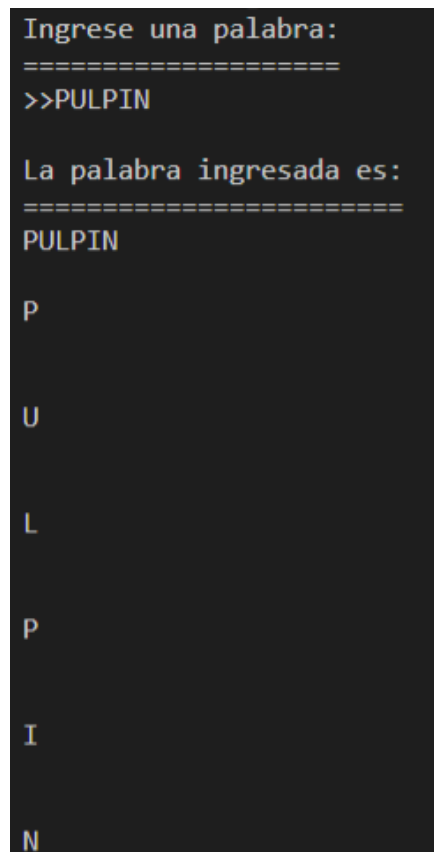
## Código – Visual Studio Code



The screenshot shows the Visual Studio Code editor with a C program named `apunadoresCadena.c`. The code is as follows:

```
1 #include <stdio.h>
2
3 /* Autor: Sánchez Estrada Angel Isaac
4  * Nacionalidad: Mexicana
5  * Fecha de elaboración: 22-03-2021
6  * Última modificación: 22-03-2021
7  * Sistema Operativo: Windows 10
8  */
9
10 /* Este programa muestra el manejo de cadenas en lenguaje C.
11 */
12 int main()
13 {
14     //Definimos a las matrices
15     char palabra[30];
16     int i=0;
17     //Impresión en pantalla para pedir que ingrese una palabra
18     printf("Ingrese una palabra:\n");
19     printf("=====\n");
20     printf(">>");
21     scanf("%s", palabra);
22
23     //Imprime la palabra que escribimos
24     printf("\nLa palabra ingresada es:\n");
25     printf("=====\n");
26     printf("%s", palabra);
27
28     //Función para guardar la pabra en la marices
29     for (i = 0 ; i < 20 ; i++)
30     {
31         printf("%c\n", palabra[i]);
32     }
33     return 0;
34 }
```

## Ejecución



The terminal window shows the execution of the program. The output is as follows:

```
Ingrese una palabra:
=====
>>PULPIN

La palabra ingresada es:
=====
PULPIN

P

U

L

P

I

N
```

## **Bibliografía:**

- Díaz, G. Apuntadores en Lenguaje C (Doctoral dissertation, Universidad de Los Andes).
- Buriticá, O. I. T. (2017). Programación imperativa con lenguaje C. Ecoe Ediciones.
- Pazos, R., Antolino, A., Pérez, J., & Sosa, V. J. (1996). Manejo de apuntadores en un Manejador de Bases de Datos Distribuidas Relacional. In Memoria Electro-Congreso Internacional de Ingeniería Electrónica (Vol. 18, pp. 38-43).

## **Referencias:**

- Laboratorio Salas A y B. (s. f.). Laboratorio de Computación Salas A y B. Recuperado el 22 de marzo del 2021, de <http://lcp02.fi-b.unam.mx>
- Apuntadores. (s. f.). webdelprofesor.ula.ve. Recuperado 22 de marzo de 2021, de [http://webdelprofesor.ula.ve/ingenieria/gilberto/pr2/01\\_Apuntadores.pdf](http://webdelprofesor.ula.ve/ingenieria/gilberto/pr2/01_Apuntadores.pdf)
- Christian León. (2014, 1 enero). Aplicación de aritmética de apuntadores - Programación en C [Vídeo]. YouTube. <https://www.youtube.com/watch?v=fYaBCgiG8xA&feature=youtu.be>