

# Laboratorios de computación salas A y B

---

*Profesor:*

Marco Antonio Martínez Quitana

---

*Asignatura:*

Principios de Programación

---

*Grupo:*

3

---

*No de Práctica(s):*

Practica 11

---

*Integrante(s):*

Daniela Cano Ramírez

---

*No. de Equipo de  
cómputo empleado:*

No aplica

---

*No. de Lista o Brigada:*

5

---

*Semestre:*

Primer Semestre

---

*Fecha de entrega:*

Lunes 04 de Enero del 2021

---

*Observaciones:*

---

---

**CALIFICACIÓN:** \_\_\_\_\_

## Guía práctica de estudio 11: Arreglos unidimensionales y multidimensionales

### ***Objetivos:***

Reconocer la importancia y utilidad de los arreglos, en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, así como trabajar con arreglos tanto unidimensionales como multidimensionales.

### ***Introducción:***

Los arreglos son una colección de elementos, una característica de estos datos que se incluyen es que tienen que ser del mismo tipo (entero, carácter, flotante, etc) y pueden tener una o varias dimensiones. Los arreglos tienen un tamaño definido y este no cambiara después de haber sido establecido, para acceder a cada elemento disponible en el arreglo se utilizan subíndices.

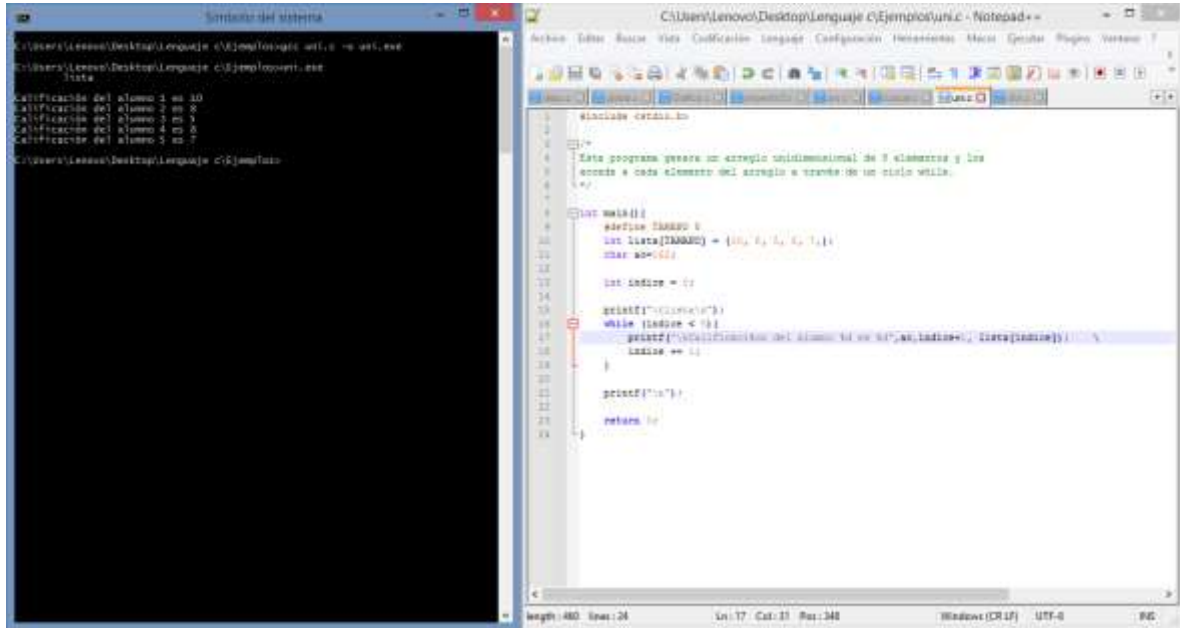
Los apuntadores son variables que almacenan direcciones de una memoria, la dirección de este apuntador corresponde al de una variable que contiene un valor en específico; resultan de utilidad para obtener un código más compacto y eficiente. En ocasiones el mal empleo de los apuntadores puede complicar el entendimiento de

### ***Actividades:***

- Elaborar un programa en lenguaje C que emplee arreglos de una dimensión.
- Resolver un problema que requiera el uso de un arreglo de dos dimensiones, a través de un programa en lenguaje C.
- Manipular arreglos a través de índices y apuntadores.

Capturas de los códigos ejemplificados en el desarrollo de la práctica:

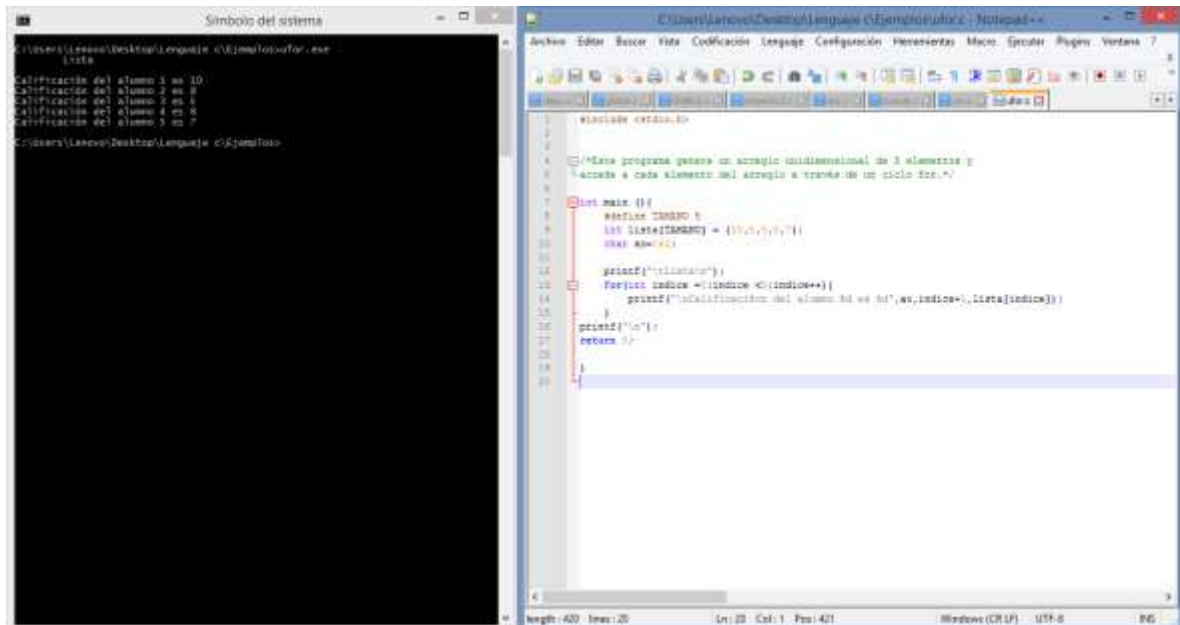
### *Código y símbolo del sistema para arreglo unidimensional while*



```
Simbolo del sistema
C:\Users\Lenovo\Desktop\Lenguaje C\Ejemplos\while > uni.exe
C:\Users\Lenovo\Desktop\Lenguaje C\Ejemplos\while.exe
Lista
Calificación del alumno 1 es 10
Calificación del alumno 2 es 8
Calificación del alumno 3 es 6
Calificación del alumno 4 es 5
Calificación del alumno 5 es 7
C:\Users\Lenovo\Desktop\Lenguaje C\Ejemplos>
```

```
C:\Users\Lenovo\Desktop\Lenguaje C\Ejemplos\while.c - Notepad++
Archivo Editar Buscar Vista Configuración Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ?
1 //Inicio programa
2
3 /*
4  *Este programa genera un arreglo unidimensional de 5 elementos y los
5  *accede a cada elemento del arreglo a través de un ciclo while.
6  */
7
8 int main()
9 {
10     #define TAMARRO 5
11     int lista[TAMARRO] = {10, 8, 6, 5, 7};
12     char nombre[50];
13
14     int indice = 0;
15
16     printf("Inicio\n");
17     while (indice < TAMARRO)
18     {
19         printf("Calificación del alumno %d es %d", indice, lista[indice]);
20         indice++;
21     }
22     printf("\n");
23     return 0;
24 }
```

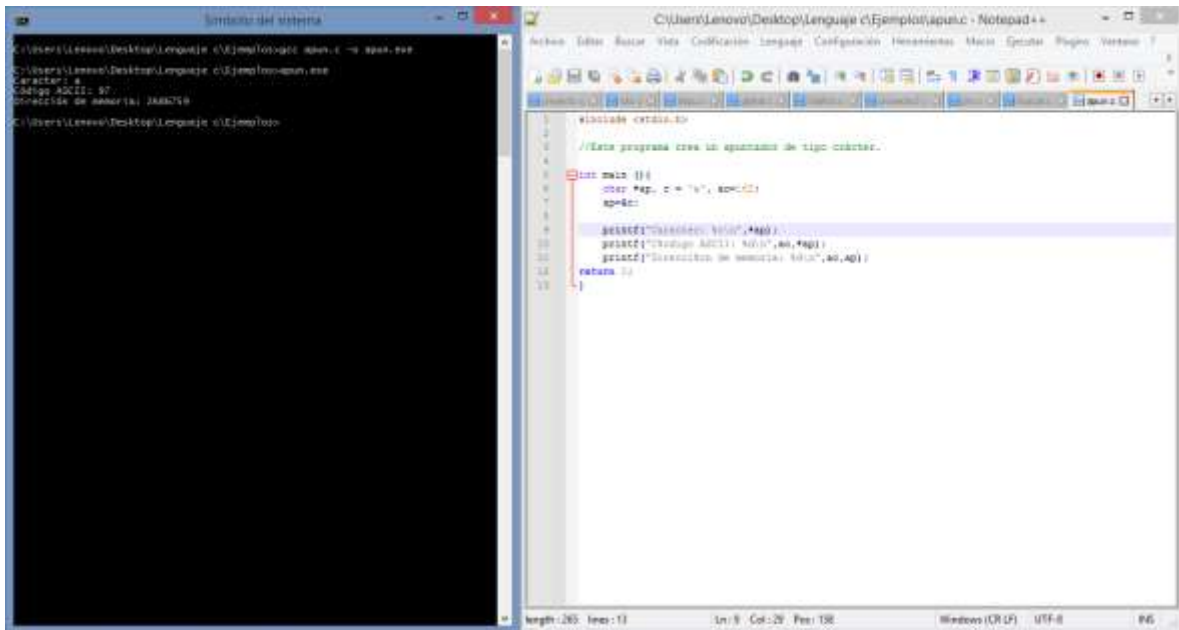
### *Código y símbolo del sistema para arreglo unidimensional for*



```
Simbolo del sistema
C:\Users\Lenovo\Desktop\Lenguaje C\Ejemplos\for.exe
Lista
Calificación del alumno 1 es 10
Calificación del alumno 2 es 8
Calificación del alumno 3 es 6
Calificación del alumno 4 es 5
Calificación del alumno 5 es 7
C:\Users\Lenovo\Desktop\Lenguaje C\Ejemplos>
```

```
C:\Users\Lenovo\Desktop\Lenguaje C\Ejemplos\for.c - Notepad++
Archivo Editar Buscar Vista Configuración Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ?
1 //Inicio programa
2
3 /*
4  *Este programa genera un arreglo unidimensional de 5 elementos y
5  *accede a cada elemento del arreglo a través de un ciclo for.
6  */
7
8 int main()
9 {
10     #define TAMARRO 5
11     int lista[TAMARRO] = {10, 8, 6, 5, 7};
12     char nombre[50];
13
14     printf("Inicio\n");
15     for (int indice = 0; indice < TAMARRO; indice++)
16     {
17         printf("Calificación del alumno %d es %d", indice, lista[indice]);
18     }
19     printf("\n");
20     return 0;
21 }
```

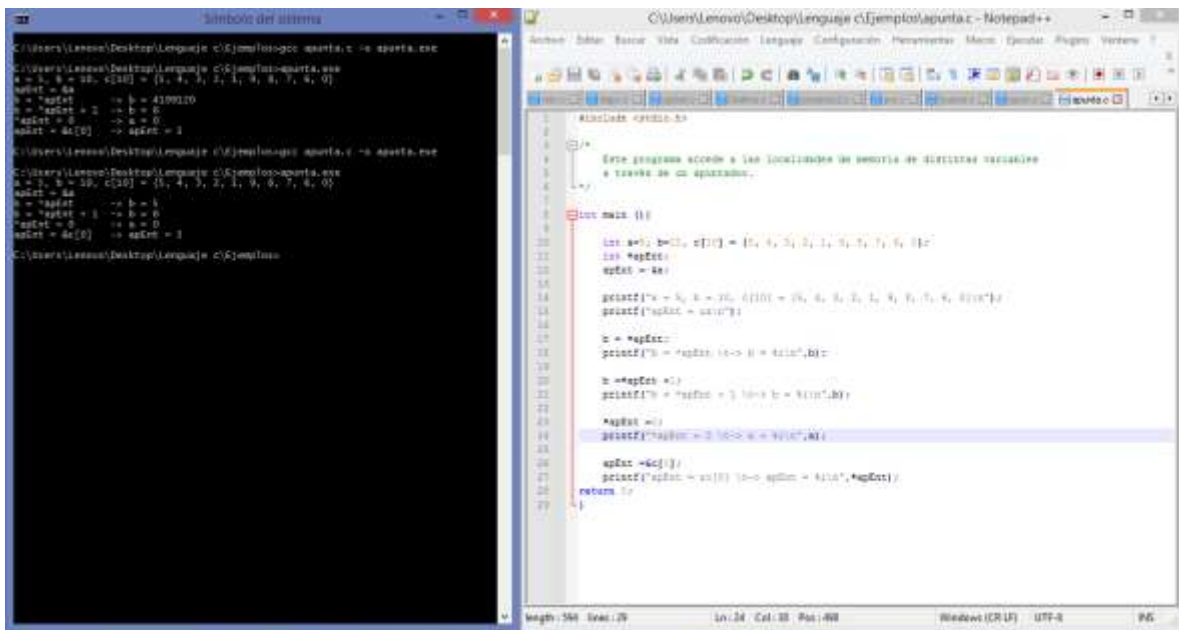
## Código y símbolo del sistema para apuntadores



```
Symbol del sistema
C:\Users\Lenovo\Desktop\Language c\Ejemplos\apunta.c -> apunta.exe
Caracter: a
Código ASCII: 97
Dirección de memoria: 2A8B7168
C:\Users\Lenovo\Desktop\Language c\Ejemplos>

C:\Users\Lenovo\Desktop\Language c\Ejemplos\apunta.c - Notepad++
Archivo Editar Buscar Vista Configuración Language Configuración Herramientas Macro Ejecutar Plugins Ventana ?
1 //Ejemplo <apunta.c>
2
3 //Este programa crea un apuntador de tipo carácter.
4
5 int main () {
6     char *ap;
7     ap = &a;
8
9     printf("Dirección: %p\n", ap);
10    printf("Contenido ASCII: %d\n", *ap);
11    printf("Dirección de memoria: %d\n", &a);
12    return 0;
13 }
```

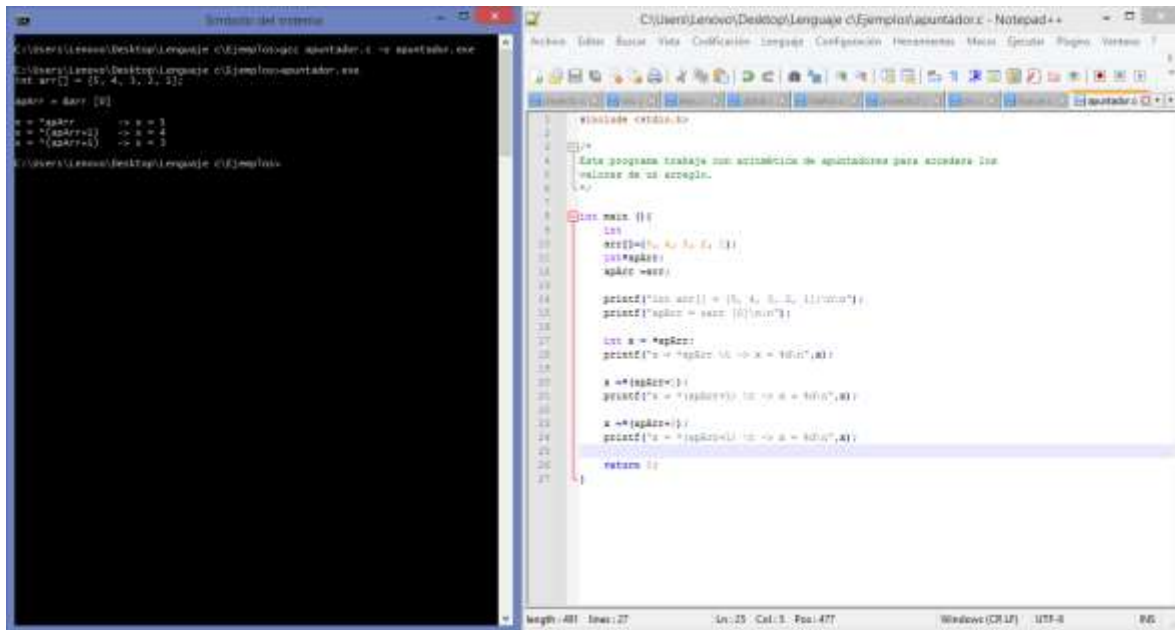
## Código y símbolo del sistema para apuntadores



```
Symbol del sistema
C:\Users\Lenovo\Desktop\Language c\Ejemplos\apunta.c -> apunta.exe
Caracter: a
Código ASCII: 97
Dirección de memoria: 2A8B7168
C:\Users\Lenovo\Desktop\Language c\Ejemplos>

C:\Users\Lenovo\Desktop\Language c\Ejemplos\apunta.c - Notepad++
Archivo Editar Buscar Vista Configuración Language Configuración Herramientas Macro Ejecutar Plugins Ventana ?
1 //Ejemplo <apunta.c>
2
3 //Este programa accede a las localizaciones de memoria de distintas variables
4 //a través de un apuntador.
5
6 int main () {
7     int a=5, b=10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0};
8     char *ap;
9     ap = &a;
10    printf("a = %d, b = %d, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0}\n");
11    printf("ap = %p\n", ap);
12
13    b = *ap;
14    printf("b = %d\n", b);
15
16    b = *ap + 1;
17    printf("b = %d\n", b);
18
19    *ap = 0;
20    printf("a = %d\n", a);
21
22    *ap = 1;
23    printf("a = %d\n", a);
24
25    return 0;
26 }
```

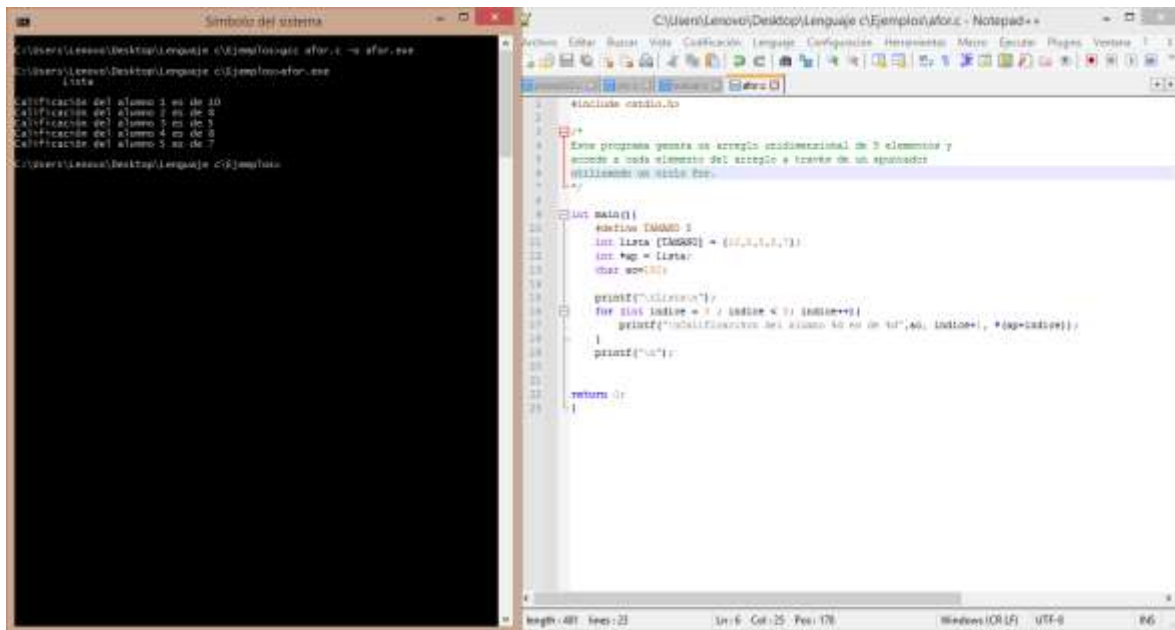
## *Código y símbolo del sistema para aritmética con apuntadores*



```
Windows (CR LF)
C:\Users\Lenovo\Desktop\lenguaje c\Ejemplo\apuntador.c -> apntador.exe
C:\Users\Lenovo\Desktop\lenguaje c\Ejemplo\apntador.exe
int arr[] = {5, 4, 3, 2, 1};
apnt = arr;
*apnt -> 5
*(apnt+2) -> 3
*(apnt+4) -> 1
C:\Users\Lenovo\Desktop\lenguaje c\Ejemplo>

C:\Users\Lenovo\Desktop\lenguaje c\Ejemplo\apntador.c - Notepad++
Archivo Editar Buscar Vista Configuración Lenguaje Configuración Herramientas Menú Ayuda Plugins Ventana
1 //include <stdio.h>
2
3 /*
4  * Este programa trabaja con aritmética de apuntadores para acceder los
5  * valores de un arreglo.
6  */
7
8 int main() {
9     int
10     arr[] = {5, 4, 3, 2, 1};
11     int *apnt;
12     apnt = arr;
13
14     printf("arr[] = {5, 4, 3, 2, 1}\n");
15     printf("apnt = arr: %i\n", arr);
16
17     int x = *apnt;
18     printf("x = *apnt %d => x = %d\n", x);
19
20     x = *(apnt+2);
21     printf("x = *(apnt+2) => x = %d\n", x);
22
23     x = *(apnt+4);
24     printf("x = *(apnt+4) => x = %d\n", x);
25
26     return 0;
27 }
```

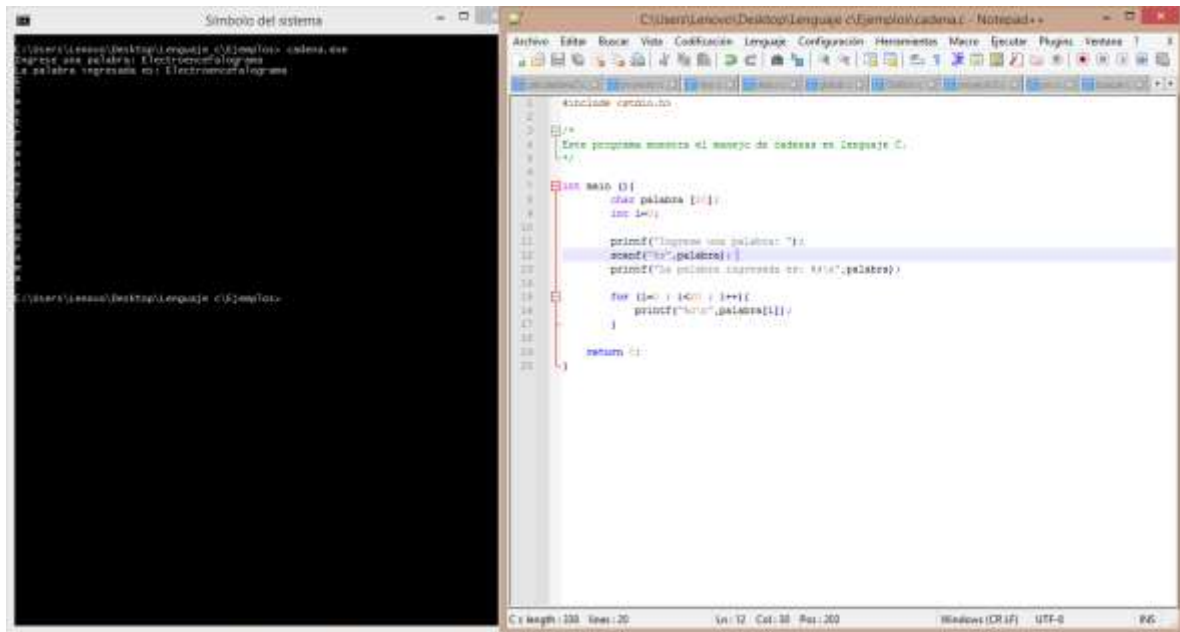
## *Código y símbolo del sistema para apuntadores en ciclo for*



```
Windows (CR LF)
C:\Users\Lenovo\Desktop\lenguaje c\Ejemplo\afor.c -> afor.exe
C:\Users\Lenovo\Desktop\lenguaje c\Ejemplo\afor.exe
Lista
Calificación del alumno 1 es de 10
Calificación del alumno 2 es de 5
Calificación del alumno 3 es de 5
Calificación del alumno 4 es de 5
Calificación del alumno 5 es de 7
C:\Users\Lenovo\Desktop\lenguaje c\Ejemplo>

C:\Users\Lenovo\Desktop\lenguaje c\Ejemplo\afor.c - Notepad++
Archivo Editar Buscar Vista Configuración Lenguaje Configuración Herramientas Menú Ayuda Plugins Ventana
1 //include <stdio.h>
2
3 /*
4  * Este programa genera un arreglo multidimensional de 5 elementos y
5  * accede a cada elemento del arreglo a través de un apuntador
6  * utilizando un ciclo for.
7  */
8
9 int main() {
10     int
11     arreglo [MAGNO 5] = {{1,2,3,4,5}};
12     int *ap;
13     char *nom;
14
15     printf("Lista\n");
16     for (int indice = 0; indice < 5; indice++) {
17         printf("Calificación del alumno %d es de %d", indice+1, *(ap+indice));
18     }
19     printf("\n");
20
21     return 0;
22 }
23 }
```

## Código y símbolo del sistema para apuntadores en cadenas



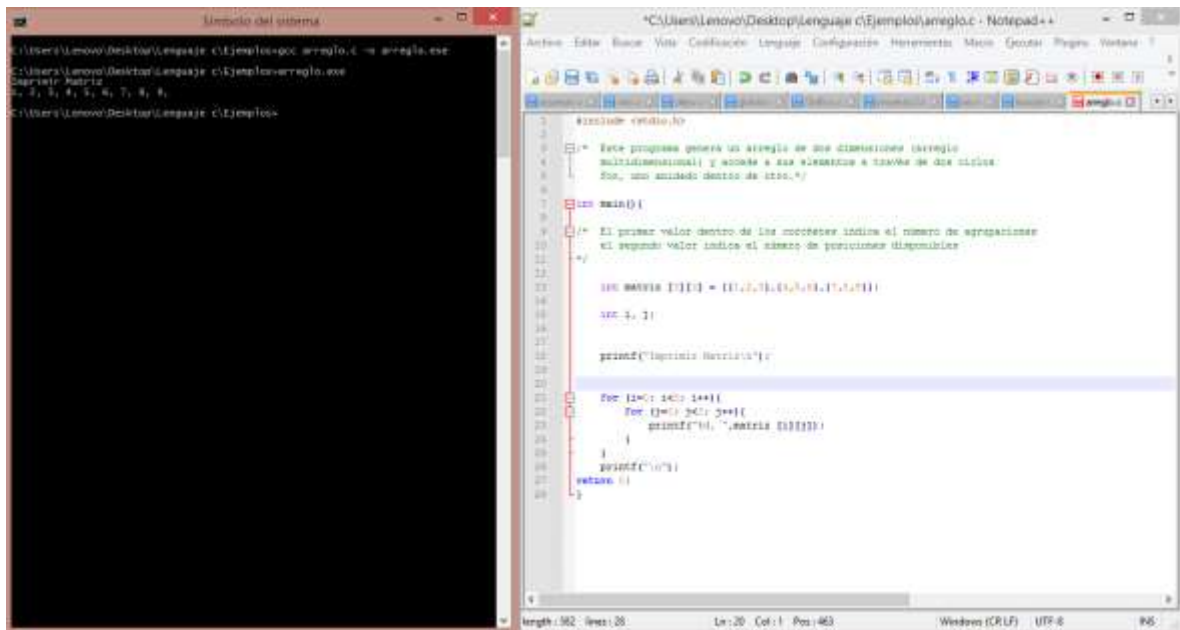
The screenshot shows a Windows command prompt window on the left and a Notepad++ editor on the right. The command prompt displays the execution of a program named 'cadena.exe' which prints 'Ingrese una palabra: Electroneca' and 'La palabra ingresada es: Electroneca'. The Notepad++ editor shows the source code for 'cadena.c'.

```
Simbolo del sistema
C:\Users\Lenovo\Desktop\lenguaje c\Ejemplos> cadena.exe
Ingrese una palabra: Electroneca
La palabra ingresada es: Electroneca

C:\Users\Lenovo\Desktop\lenguaje c\Ejemplos>
```

```
C:\Users\Lenovo\Desktop\lenguaje c\Ejemplos\cadena.c - Notepad++
Archivo Editar Buscar Vista Configuración Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana 1
1 #include <stdio.h>
2
3 /*
4  * Este programa muestra el manejo de cadenas en lenguaje C.
5  */
6
7 int main ()
8 {
9     char palabra [50];
10    int i=0;
11
12    printf("Ingrese una palabra: ");
13    scanf("%s",palabra);
14    printf("La palabra ingresada es: %s",palabra);
15
16    for (i=0; i<strlen(palabra); i++)
17    {
18        printf("%c",palabra[i]);
19    }
20
21    return 0;
22 }
```

## Código y símbolo del sistema para arreglos multidimensionales



The screenshot shows a Windows command prompt window on the left and a Notepad++ editor on the right. The command prompt displays the execution of a program named 'arreglo.c' which prints a 2x3 matrix of numbers. The Notepad++ editor shows the source code for 'arreglo.c'.

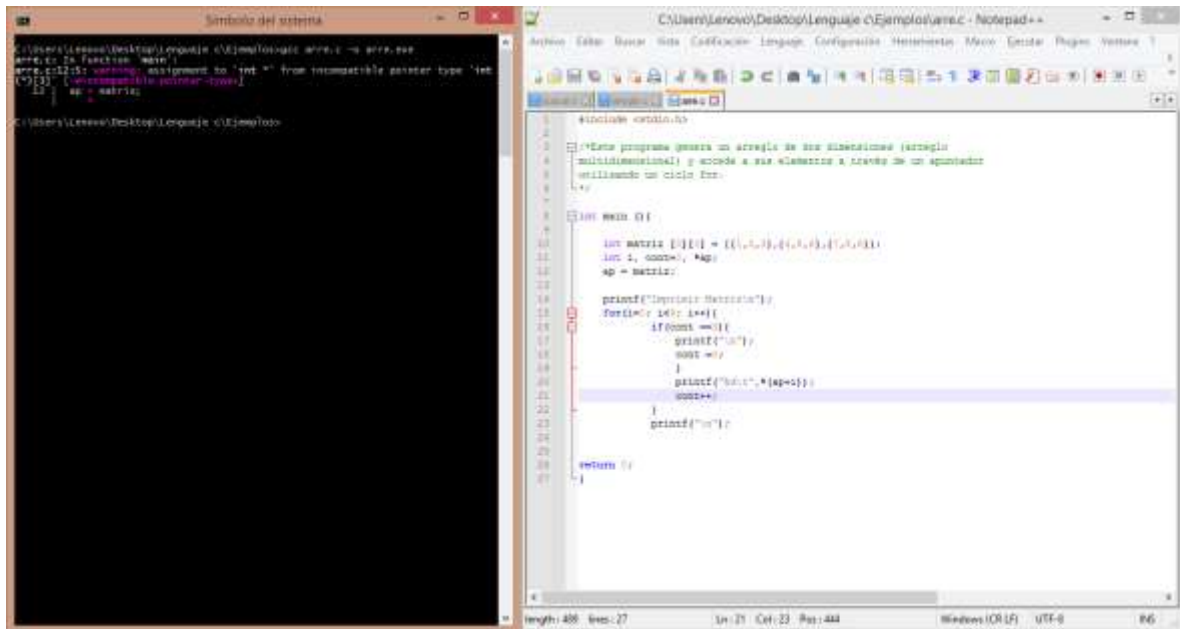
```
Simbolo del sistema
C:\Users\Lenovo\Desktop\lenguaje c\Ejemplos> arreglo.c - arreglo.exe
C:\Users\Lenovo\Desktop\lenguaje c\Ejemplos> arreglo.exe
Matriz
1 2 3
4 5 6
7 8 9

C:\Users\Lenovo\Desktop\lenguaje c\Ejemplos>
```

```
C:\Users\Lenovo\Desktop\lenguaje c\Ejemplos\arreglo.c - Notepad++
Archivo Editar Buscar Vista Configuración Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana 1
1 #include <stdio.h>
2
3 /*
4  * Este programa genera un arreglo de dos dimensiones (arreglo
5  * multidimensional) y recorre a sus elementos a través de dos ciclos.
6  * Son, uno anidado dentro de otro.*/
7
8 int main ()
9 {
10    /* El primer valor dentro de los corchetes indica el número de agrupaciones
11    * el segundo valor indica el número de posiciones disponibles
12    */
13
14    int matriz [3][3] = {{1,2,3},{4,5,6},{7,8,9}};
15
16    int i, j;
17
18    printf("Imprima Matriz");
19
20
21    for (i=0; i<3; i++)
22    {
23        for (j=0; j<3; j++)
24        {
25            printf("%d",matriz [i][j]);
26        }
27        printf("\n");
28    }
29
30    return 0;
31 }
```

## *Código y símbolo del sistema para arreglos multidimensionales con apuntadores*

En el caso de este ejemplo de código al momento de querer compilarlo me saltaba un error.



```
Símbolo del sistema
C:\Users\Lenovo\Desktop\Lenguaje C\Ejemplos> gcc.c -o arr.c.exe
arr.c:12:13: error: variable assignment to 'int*' from incompatible pointer type 'int (*)[3]'
12 |     ap = matriz;
    |     ^
C:\Users\Lenovo\Desktop\Lenguaje C\Ejemplos>
```

```
C:\Users\Lenovo\Desktop\Lenguaje C\Ejemplos> gcc.c -o arr.c.exe
arr.c:12:13: error: variable assignment to 'int*' from incompatible pointer type 'int (*)[3]'
12 |     ap = matriz;
    |     ^
C:\Users\Lenovo\Desktop\Lenguaje C\Ejemplos>
```

```
#include <stdio.h>

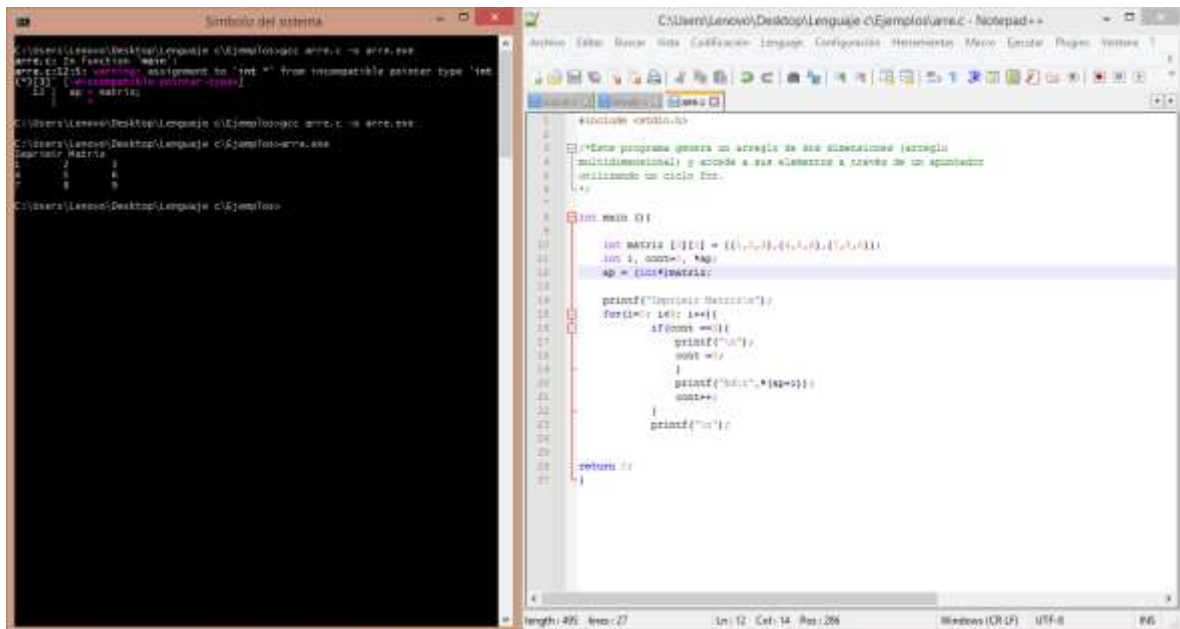
/*Este programa genera un arreglo de dos dimensiones (arreglo
multidimensional) y accede a sus elementos a través de un apuntador
utilizando un ciclo for.
*/

int main ()
{
    int matriz [3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, count=0, *ap;
    ap = matriz;

    printf("Imprimir Matriz\n");
    for(i=0; i<3; i++){
        if(count==0){
            printf("\n");
            count++;
        }
        printf("%d\t",*(ap+i));
        count++;
    }
    printf("\n");

    return 0;
}
```

Debido a eso se buscó información y le agregue al código (int\*) en la línea 12 y el programa se pudo ejecutar



```
Símbolo del sistema
C:\Users\Lenovo\Desktop\Lenguaje C\Ejemplos> gcc.c -o arr.c.exe
arr.c:12:13: error: variable assignment to 'int*' from incompatible pointer type 'int (*)[3]'
12 |     ap = matriz;
    |     ^
C:\Users\Lenovo\Desktop\Lenguaje C\Ejemplos>
```

```
C:\Users\Lenovo\Desktop\Lenguaje C\Ejemplos> gcc.c -o arr.c.exe
arr.c:12:13: error: variable assignment to 'int*' from incompatible pointer type 'int (*)[3]'
12 |     ap = matriz;
    |     ^
C:\Users\Lenovo\Desktop\Lenguaje C\Ejemplos>
```

```
#include <stdio.h>

/*Este programa genera un arreglo de dos dimensiones (arreglo
multidimensional) y accede a sus elementos a través de un apuntador
utilizando un ciclo for.
*/

int main ()
{
    int matriz [3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, count=0, *ap;
    ap = (int*)matriz;

    printf("Imprimir Matriz\n");
    for(i=0; i<3; i++){
        if(count==0){
            printf("\n");
            count++;
        }
        printf("%d\t",*(ap+i));
        count++;
    }
    printf("\n");

    return 0;
}
```

### ***Conclusiones:***

Los índices en los arreglos comienzan con el número cero por lo que si se tiene un arreglo de 10 elementos los índices comenzaran en el número 0 y terminan en el 9.

Con los arreglos es posible disminuir el tamaño del programa ya que se pueden agrupar variables, esto puede ser útil en listados (fechas, calificaciones, etc).

Los apuntadores permiten tener un mayor control con respecto a las variables que se utilizan en el programa.

### ***Referencias***

Gómez, R. (s. f.). Los Apuntadores. Recuperado 6 de enero de 2021, de

<http://cryptomex.org/Tutorial-LengC/apuntadores.html>

Laboratorio de Computación Salas A y B. Facultad de Ingeniería (2018, abril 6). Manual de prácticas Fundamento de Programación. Recuperado 04 de Enero de 2021, de

<http://lcp02.fi-b.unam.mx/>

M. (2013, 18 febrero). Pointer types, pointer arithmetic, void pointers. YouTube.

<https://www.youtube.com/watch?v=JTttg85xsbo&feature=youtu.be>