



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

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

*Asignatura:* Fundamentos de Programación

*Grupo:* 3

*No de Práctica(s):* #09

*Integrante(s):* Cuevas Antunez Samantha

*No. de Equipo de  
cómputo empleado:* No aplica

*No. de Lista o Brigada:* 12

*Semestre:* Primer semestre

*Fecha de entrega:* 30/11/2020

*Observaciones:*

CALIFICACIÓN: \_\_\_\_\_



# Estructuras de repetición



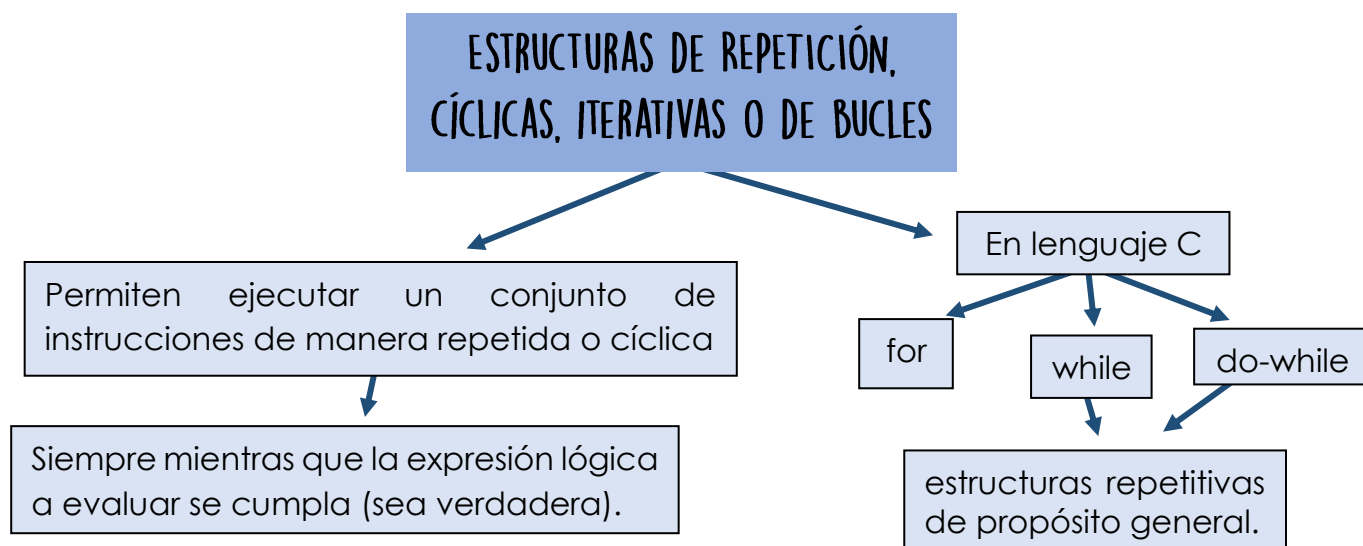
## OBJETIVOS

Elaborar programas en C para la resolución de problemas básicos que incluyan las estructuras de repetición y la directiva define.

## ACTIVIDADES

- Elaborar un programa que utilice la estructura while en la solución de un problema
- Elaborar un programa que requiera el uso de la estructura do-while para resolver un problema. Hacer la comparación con el programa anterior para distinguir las diferencias de operación entre while y do-while.
- Resolver un problema dado por el profesor que utilice la estructura for en lugar de la estructura while.
- Usar la directiva define para elaboración de código versátil.

## INTRODUCCIÓN



## ESCTRUC- TURA

## SINTAXIS

### WHILE

```
while (expresión lógica)
{
    // instrucciones a
    repetir
}
```

Primero valida la expresión lógica y si ésta es verdadera procede a ejecutar el bloque de instrucciones de la estructura, el cual está delimitado por las llaves {}. \*Si el bloque de código a repetir consta de una sola sentencia, entonces se pueden omitir las llaves.

Si la condición no se cumple se continúa el flujo normal del programa sin ejecutar el bloque de la estructura.

### DO WHILE

```
do
{
    // instrucciones a
    repetir
} while (condición o
condiciones);
```

Estructura cíclica que ejecuta el bloque de código que se encuentra dentro de las llaves y después valida la condición, es decir, el bloque de código se ejecuta de una a ene veces

### FOR

```
for (inicialización;
expresión_lógica;
operaciones por
iteración)
{
    //instrucciones a
    repetir
}
```

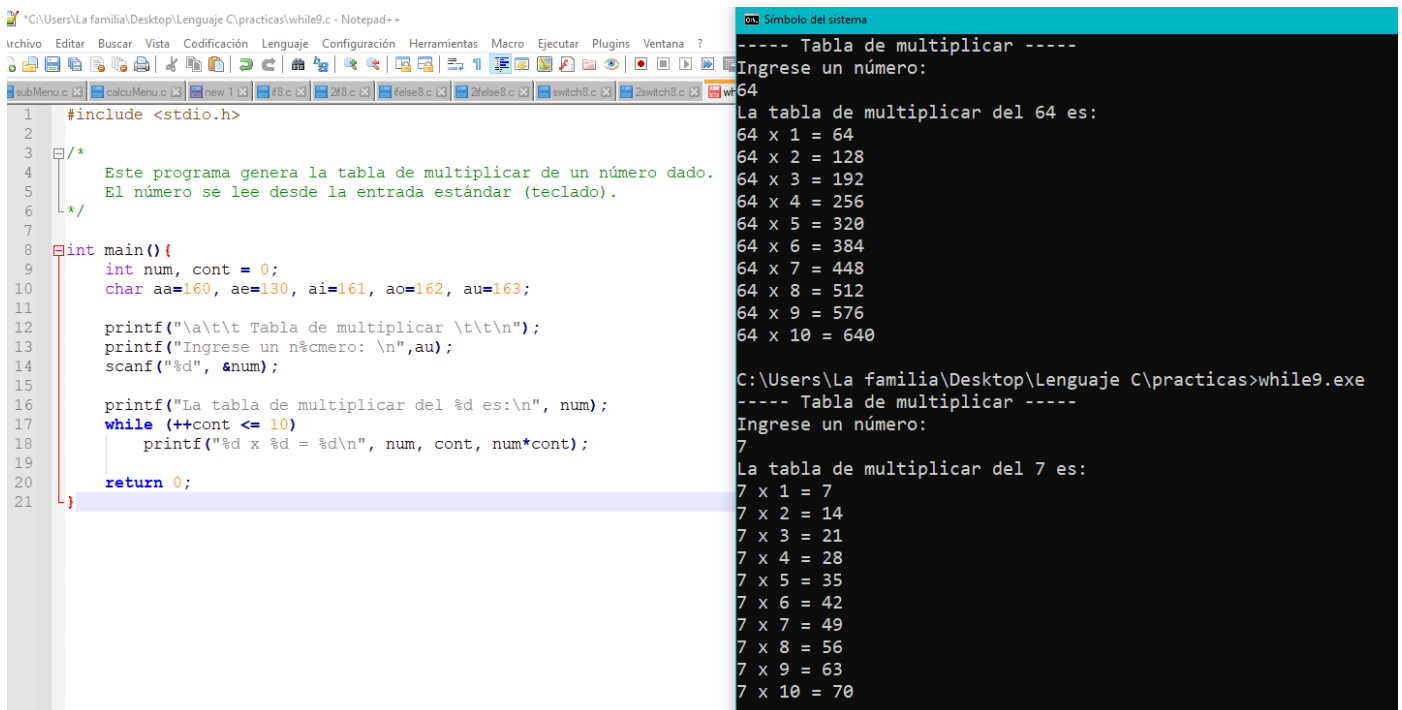
Realiza repeticiones cuando se conoce el número de elementos que se quiere recorrer La estructura for ejecuta 3 acciones básicas antes o después de ejecutar el bloque de código:

La primera acción es la inicialización, en la cual se pueden definir variables e inicializar sus valores; esta parte solo se ejecuta una vez cuando se ingresa al ciclo y es opcional. La segunda acción consta de una expresión lógica, la cual se evalúa y, si ésta es verdadera, ejecuta el bloque de código, si no se cumple se continúa la ejecución del programa; esta parte es opcional.

La tercera parte consta de un conjunto de operaciones que se realizan cada vez que termina de ejecutarse el bloque de código y antes de volver a validar la expresión lógica; esta parte también es opcional.

## RESULTADOS

### Ejemplos. WHILE



The image shows a C program in Notepad++ and its execution in a command prompt. The program uses a while loop to generate multiplication tables for a given number. It prompts the user to enter a number and then displays the multiplication table for that number, showing products from 1 to 10.

```
#include <stdio.h>

/*
 * Este programa genera la tabla de multiplicar de un número dado.
 * El número se lee desde la entrada estándar (teclado).
 */

int main() {
    int num, cont = 0;
    char aa=160, ae=130, ai=161, ao=162, au=163;

    printf("\a\t\t Tabla de multiplicar \t\t\t\n");
    printf("Ingrese un número: \n", au);
    scanf("%d", &num);

    printf("La tabla de multiplicar del %d es:\n", num);
    while (++cont <= 10)
        printf("%d x %d = %d\n", num, cont, num*cont);

    return 0;
}
```

Simbolo del sistema

```
----- Tabla de multiplicar -----
Ingrese un número:
64
La tabla de multiplicar del 64 es:
64 x 1 = 64
64 x 2 = 128
64 x 3 = 192
64 x 4 = 256
64 x 5 = 320
64 x 6 = 384
64 x 7 = 448
64 x 8 = 512
64 x 9 = 576
64 x 10 = 640

C:\Users\La familia\Desktop\Lenguaje C\practicass>while9.exe
----- Tabla de multiplicar -----
Ingrese un número:
7
La tabla de multiplicar del 7 es:
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
```

Imagen 1

```
chivo  Editor  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?
[Menu] [new] [3 c] [2 f8 c] [false8 c] [switch8 c] [2 switch8 c] [while9 c] [2 while9 c]

1  #include <stdio.h>
2  /*
3     Este programa genera un ciclo infinito.
4  */
5
6  int main(){
7      // Al igual que en la estructura if-else
8      // 0 -> falso
9      // diferente de 0 -> verdadero
10
11     // El siguiente es un ciclo infinito
12     // porque la condición siempre es verdadera.
13     // Así mismo, debido a que el ciclo consta de una sola línea, las
14     // llaves { } son opcionales.
15
16     while (100)
17     {
18         printf("Ciclo infinito.\nPara terminar el ciclo presione ctrl + c.\n");
19     }
20
21     return 0;
22 }
```

## Calculadora.

```

1 #include <stdio.h>
2 /* Este programa genera una calculadora básica. */
3
4 int main ()
5 {
6     int op, uno, dos;
7     char aa=160, ae=130, ai=161, ao=162, au=163, sp=168;
8     do
9     {
10         printf(" --- Calculadora ---\n");
11         printf("\n¿Qué desea hacer?\n",sp,ae);
12         printf("1) Sumar\n");
13         printf("2) Restar\n");
14         printf("3) Multiplicar\n");
15         printf("4) Dividir\n");
16         printf("5) Salir\n");
17         scanf("%d",&op);
18
19         switch(op)
20         {
21             case 1:
22                 printf("\tSumar\n");
23                 printf("Introduzca los números a sumar separados por comas\n",au);
24                 scanf("%d,%d",&uno,&dos);
25                 printf("%d + %d = %d\n", uno, dos, (uno + dos));
26                 break;
27             case 2:
28                 printf("\tRestar\n");
29                 printf("Introduzca los números a restar separados por comas\n",au);
30                 scanf("%d,%d",&uno,&dos);
31                 printf("%d - %d = %d\n", uno, dos, (uno - dos));
32                 break;
33             case 3:
34                 printf("\tMultiplicar\n");
35                 printf("Introduzca los números a multiplicar separados por comas\n",au);
36                 scanf("%d,%d",&uno,&dos);
37                 printf("%d * %d = %d\n", uno, dos, (uno * dos));
38                 break;
39             case 4:
40                 printf("\tDividir\n");
41                 printf("Introduzca los números a dividir separados por comas\n",au);
42                 scanf("%d,%d",&uno,&dos);
43                 printf("%d / %d = %.21f\n", uno, dos, ((double)uno / dos));
44                 break;
45             case 5:
46                 printf("\tSalir\n");
47                 break;
48             default:
49                 printf("\tOpción inválida.\n",ao,aa);
50         }
51     } while (op != 5);
52     return 0;
53 }
54
55
56

```

Imagen 5

```

C:\Users\La familia\Desktop\Lenguaje C\practicac>gcc 2dowhile9.c
C:\Users\La familia\Desktop\Lenguaje C\practicac>2dowhile9.exe

--- Calculadora ---

¿Qué desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
1
    Sumar
Introduzca los números a sumar separados por comas
2,64
2 + 64 = 66
--- Calculadora ---

¿Qué desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
2
    Restar
Introduzca los números a restar separados por comas
9,7
9 - 7 = 2
--- Calculadora ---

9,7
9 - 7 = 2
--- Calculadora ---

¿Qué desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
3
    Multiplicar
Introduzca los números a multiplicar separados por comas
9,47
9 * 47 = 423
--- Calculadora ---

¿Qué desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
4
    Dividir
Introduzca los números a dividir separados por comas
74,6
74 / 6 = 12.33
--- Calculadora ---

¿Qué desea hacer

```

Imagen 6

```

Introduzca los números a dividir separados por comas
74,6
74 / 6 = 12.33
--- Calculadora ---

¿Qué desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
9
    Opción inválida.
--- Calculadora ---

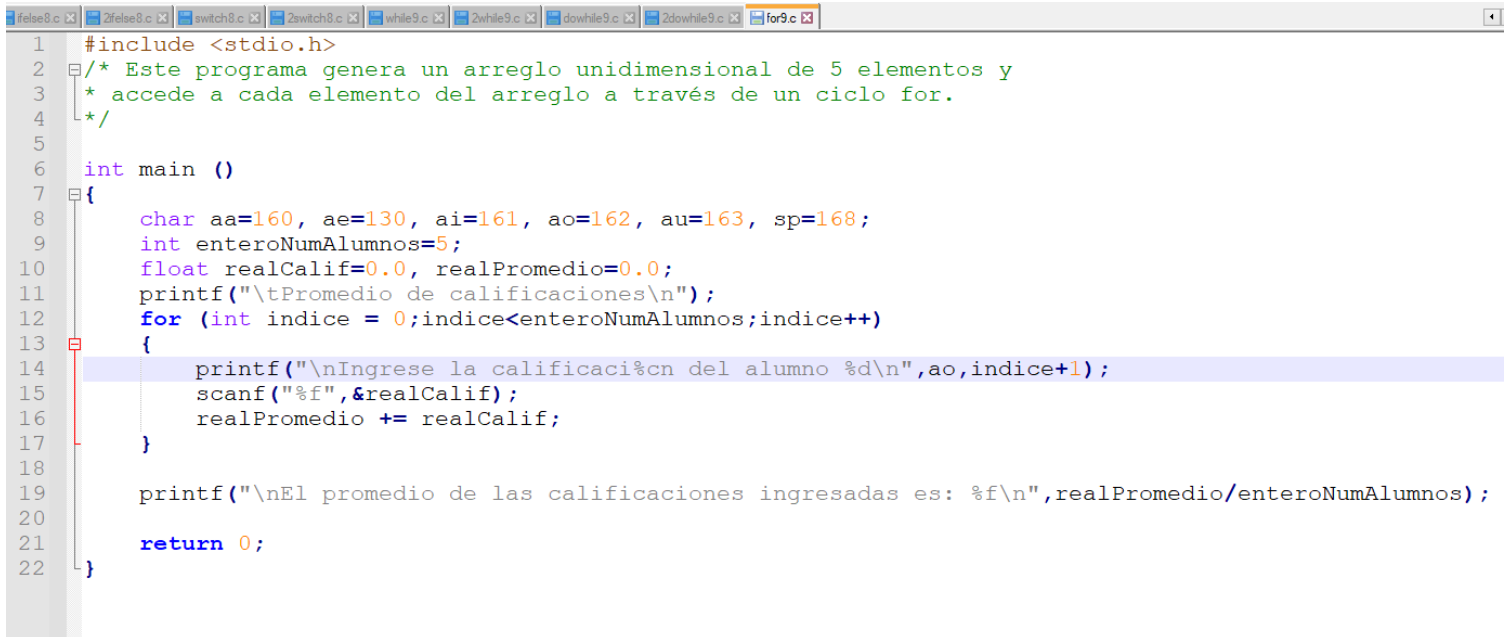
¿Qué desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
5
    Salir

C:\Users\La familia\Desktop\Lenguaje C\practicac>

```

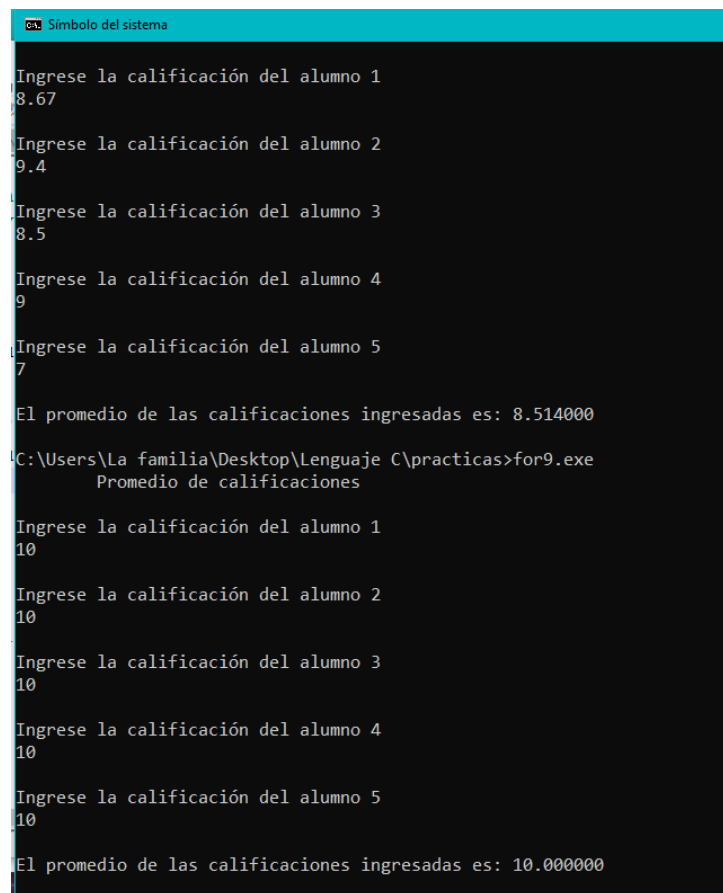
Imagen 7

## Ejemplos FOR



```
1 #include <stdio.h>
2 /* Este programa genera un arreglo unidimensional de 5 elementos y
3  * accede a cada elemento del arreglo a través de un ciclo for.
4  */
5
6 int main ()
7 {
8     char aa=160, ae=130, ai=161, ao=162, au=163, sp=168;
9     int enteroNumAlumnos=5;
10    float realCalif=0.0, realPromedio=0.0;
11    printf("\tPromedio de calificaciones\n");
12    for (int indice = 0; indice<enteroNumAlumnos; indice++)
13    {
14        printf("\nIngrese la calificación del alumno %d\n",ao,indice+1);
15        scanf("%f",&realCalif);
16        realPromedio += realCalif;
17    }
18
19    printf("\nEl promedio de las calificaciones ingresadas es: %f\n",realPromedio/enteroNumAlumnos);
20
21    return 0;
22 }
```

Imagen 8



```
Simbolo del sistema
Ingrese la calificación del alumno 1
8.67

Ingrese la calificación del alumno 2
9.4

Ingrese la calificación del alumno 3
8.5

Ingrese la calificación del alumno 4
9

Ingrese la calificación del alumno 5
7

El promedio de las calificaciones ingresadas es: 8.514000
C:\Users\La familia\Desktop\Lenguaje C\practicass>for9.exe
Promedio de calificaciones

Ingrese la calificación del alumno 1
10

Ingrese la calificación del alumno 2
10

Ingrese la calificación del alumno 3
10

Ingrese la calificación del alumno 4
10

Ingrese la calificación del alumno 5
10

El promedio de las calificaciones ingresadas es: 10.000000
```

Imagen 9

## CONCLUSIONES

Las estructuras de repetición, como su nombre lo dicen, repiten 1 o más veces ciertas instrucciones “n” número determinado de veces o mientras se cumpla la condición.

While, valida la condición o condiciones y la ejecuta hasta que esta ya no sea verdadera, en el segundo ejemplo del while el ciclo se siguió infinitamente debido a que la condición siempre fue verdadera.

La estructura do-while es muy en los menús, en esta se realiza primero una cosa y luego se repite si se cumple la condición, una característica importante es que siempre se ejecuta al menos una vez. En los ejemplos realizados se ve más explícito esto, por ejemplo, en el promedio de calificaciones (imagen 3) nos da la pregunta si deseamos sumar otra calificación repetidamente (instrucciones del do) y al momento de seleccionar que no, corre las instrucciones del while. De igual forma en la calculadora para que no se salga del programa al hacer una operación nos deja dentro de ésta para poder volver a elegir otra operación y deja de hacer esto cuando se elige la opción de salir.

En el caso del ciclo for, repite las sentencias cuando se conoce el número de elementos que se quiere recorrer, algo que la distingue de las anteriores, además de que en esta estructura se puede definir la variable dentro de esta y no en un inicio como siempre.

A pesar de que las estructuras difieren un poco en sus características todas nos ayudan a que se repita una acción dependiendo el caso, el ejemplo más claro para mi es el ejemplo de la calculadora usando el ciclo do while, ya que gracias a este n tenemos que correr el programa a cada rato después de hacer una operación.

## REFERENCIAS

Facultad de Ingeniería. (2018, 6 abril). Manual de prácticas de fundamentos de programación MADO-17\_FP. Laboratorio de Computación Salas A y B. <http://lcp02.fi-b.unam.mx/>



