



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

Laboratorio de Computación

Salas A y B

Profesor(a): _____ Karina García Morales

Asignatura: _____ Fundamentos de Programación

Grupo: _____ 22

No de Práctica(s): _____ 08

Integrante(s): _____ Edilberto Vicente Martínez

*No. De lista o
brigada:* _____ 50

Semestre: _____ 2026-1

Fecha de entrega: _____ 28 de octubre de 2025

Observaciones: _____

CALIFICACIÓN: _____

Práctica 08: Estructuras de repetición

Objetivo:

El alumno elaborará programas en C para la resolución de problemas básicos que incluyan las estructuras de repetición.

Desarrollo:

Las estructuras de repetición son las llamadas también estructuras cíclicas, iterativas o de bucles. Permiten ejecutar un conjunto de instrucciones de manera repetida (o cíclica) mientras que la expresión lógica a evaluar se cumpla (sea verdadera). En lenguaje C existen tres estructuras de repetición: while, do-while y for. Las estructuras while y do-while son estructuras repetitivas de propósito general.

Estructura de control repetitiva while

Esta estructura primeramente valida la expresión lógica y si este cumple (es verdadera) realiza las instrucciones que están dentro de las llaves ({}) al terminar las instrucciones vuelve a ejecutar las instrucciones, siempre y cuando la expresión lógica sea verdadera. Cuando la condición no se cumple (no es verdadera) el programa sigue su flujo normal pero no realiza las instrucciones almacenadas en las llaves.

La sintaxis de la estructura while es la siguiente:

```
while (expresión_lógica) {  
    // Bloque de código a repetir  
    // mientras que la expresión  
    // lógica sea verdadera.  
}
```

Si el bloque de código a repetir consta de una sola sentencia, entonces se pueden omitir las llaves.

Estructura de control repetitiva do-while

La estructura do while es una estructura cíclica el cual primero ejecuta las instrucciones que se encuentran dentro de las llaves y después valida la condición, es decir primero se ejecuta una vez la acción y si la condición es válida se sigue repitiendo la acción una y otra vez. Su sintaxis es la siguiente:

```
do {  
/*  
Bloque de código que se ejecuta  
por lo menos una vez y se repite
```

mientras la expresión lógica sea
verdadera.

*/

} **while** (expresión_lógica);

Si el bloque de código a repetir consta de una sola sentencia, entonces se pueden omitir las llaves. Esta estructura de control siempre termina con el signo de puntuación punto y coma (;).

Estructura de control de repetición for

La estructura for permite realizar repeticiones cuando el control de la repetición está definido sobre una variable llamada contador. La sintaxis que generalmente se usa es la siguiente:

```
for (inicialización ; expresión_lógica ; operaciones por iteración) {  
    /*  
     * Bloque de código  
     * a ejecutar  
     */  
}
```

La estructura for ejecuta 3 acciones básicas, dos antes y una 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 acción 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 acción es opcional. La tercera acción 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 acción también es opcional.

Define

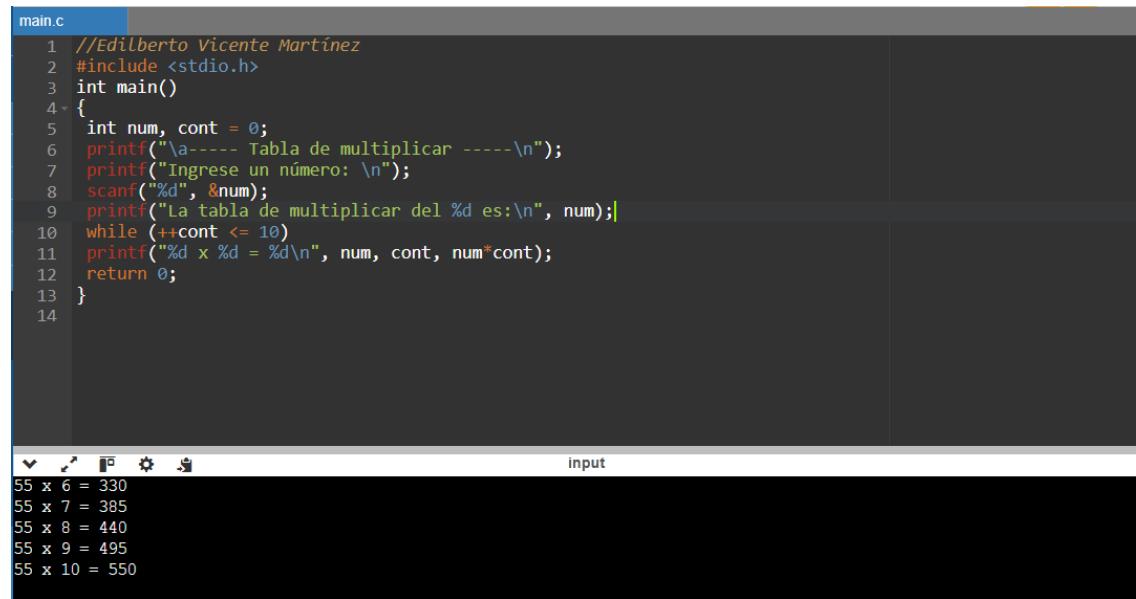
Las líneas de código que empiezan con # son directivas del preprocesador, el cual se encarga de realizar modificaciones en el texto del código fuente, como reemplazar un símbolo definido con #define por un parámetro o texto, o incluir un archivo en otro archivo con #include. define permite definir constantes o literales; se les nombra también como constantes simbólicas. Su sintaxis es la siguiente:

#define <nombree> <valor>

Al definir la constante simbólica con #define, se emplea un nombre y un valor. Cada vez que aparezca el nombre en el programa se cambiará por el valor definido. El valor puede ser numérico o puede ser texto.

1.- Revisar el uso de ++cont<=10 en el programa1.c

En este programa el ++cont nos sirve para que al correr nuestro programa el último al que se multiplique sea el número 10, ya que si ponemos el cont++ el ultimo numero a multiplicar sería el 11.

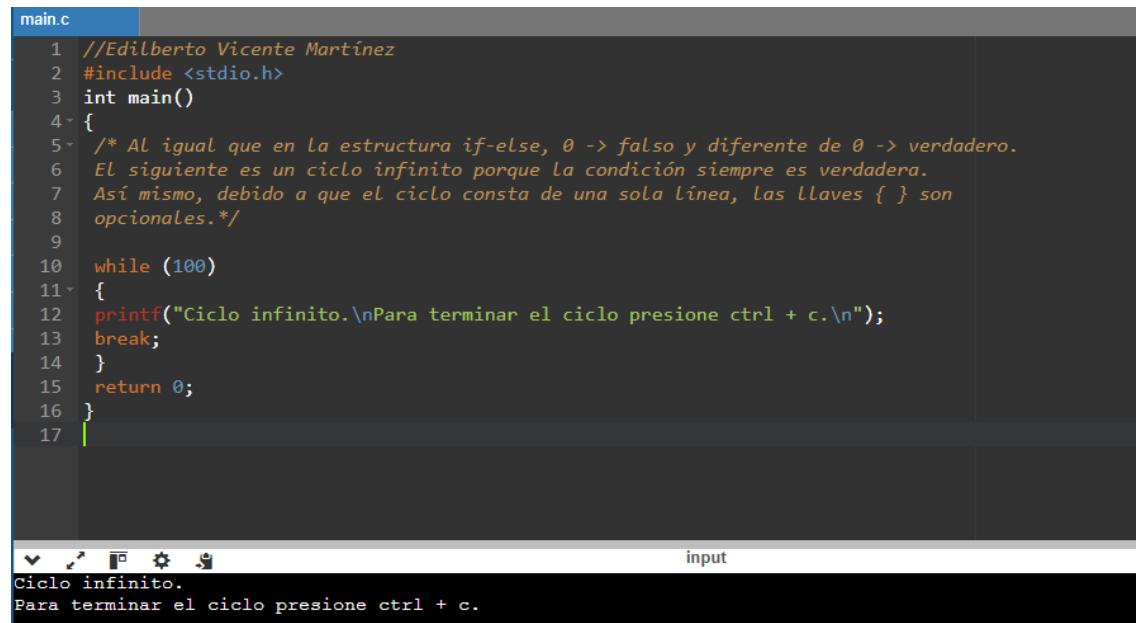


```
main.c
1 //Edilberto Vicente Martínez
2 #include <stdio.h>
3 int main()
4 {
5     int num, cont = 0;
6     printf("\n----- Tabla de multiplicar -----");
7     printf("Ingrese un número: \n");
8     scanf("%d", &num);
9     printf("La tabla de multiplicar del %d es:\n", num);
10    while (++cont <= 10)
11        printf("%d x %d = %d\n", num, cont, num*cont);
12    return 0;
13 }
```

55 x 6 = 330
55 x 7 = 385
55 x 8 = 440
55 x 9 = 495
55 x 10 = 550

*En este programa se realiza la tabla de multiplicación del 1 al 10 de cualquier número entero que ingrese el usuario.

2.- El uso de un break en el programa2.c



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

Ciclo infinito.
Para terminar el ciclo presione ctrl + c.

*Al poner el break el ciclo se termina automáticamente y solamente com un simple texto.

3.- Para el programa3.c el uso de la limpieza de cache en teclado

En este caso la limpieza ayuda a limpiar el valor anterior que puso el usuario, para asi poder pedir el otro número y si en caso se pone N, toma en cuenta los datos anteriormente ingresados.

```
main.c
1 //Edilberto Vicente Martínez
2 #include <stdio.h>
3 int main ()
4 {
5     char op = 'n';
6     double sum = 0, calif = 0;
7     int veces = 0;
8     do
9     {
10        printf("\tSuma de calificaciones\n");
11        printf("Ingrese la calificación:\n");
12        scanf("%lf", &calif);
13        veces++;
14        sum = sum + calif;
15        printf("Desea sumar otra? S/N\n");
16        setbuf(stdin, NULL); // Limpia el buffer del teclado
17        scanf("%c", &op);
18        getchar();
19    }
20    while (op == 'S' || op == 's');

```

Ingresé la calificación:
6
Desea sumar otra? S/N
n
El promedio de las calificaciones ingresadas es: 7.000000

*El programa calcula el promedio del numero de calificaciones ingresadas por el usuario.

4.-¿Para el programa4.c que pasa si divido entre cero? Deberán agregar la condición, ustedes eligen cual utilizar.

Al no tener la condición la división da como resultado un -nan, mientras que al tener la condicion no aparece el resultado de la división, solamente aparece de nuevo la opción de colocar los números a dividir.

```
main.c
28 printf("%d - %d = %d\n", uno, dos, (uno - dos));
29 break;
30 case 3:
31 printf("\tMultiplicar\n");
32 printf("Introduzca los números a multiplicar separados por comas\n");
33 scanf("%d, %d", &uno, &dos);
34 printf("%d * %d = %d\n", uno, dos, (uno * dos));
35 break;
36 case 4:
37     printf("\tDividir\n");
38     printf("Introduzca los números a dividir separados por comas\n");
39     scanf("%d, %d", &uno, &dos);
40     printf("%d / %d = %.2lf\n", uno, dos, ((double)uno / dos));
41 break;
42 case 5:
43 printf("\tSalir\n");
44 break;
45 default:
46 printf("\tOpción inválida.\n");
47 }
48 }
```

0,0
0 / 0 = -nan
--- calculadora ---

*Programa sin la condición, el resultado de 0/0 es igual a -nan

```
main.c
28 printf("%d - %d = %d\n", uno, dos, (uno - dos));
29 break;
30 case 3:
31 printf("\tMultiplicar\n");
32 printf("Introduzca los números a multiplicar separados por comas\n");
33 scanf("%d, %d", &uno, &dos);
34 printf("%d * %d = %d\n", uno, dos, (uno * dos));
35 break;
36 case 4:
37     printf("\tDividir\n");
38     do{
39         printf("Introduzca los números a dividir separados por comas\n");
40         scanf("%d, %d", &uno, &dos);
41     }while(dos==0);
42     printf("%d / %d = %.2lf\n", uno, dos, ((double)uno / dos));
43 break;
44 case 5:
45     printf("\tSalir\n");
46 break;
47 default:
48     printf("\tOpción inválida.\n");
```

input
5) Salir
4 Dividir
Introduzca los números a dividir separados por comas
0,0
Introduzca los números a dividir separados por comas

*Programa con la condición y la estructura do while, aquí el resultado de la división no aparece, solamente aparece el volver a introducir los numeros a dividir.

5.- Modifica el programa5.c Utilizando la estructura while

```
main.c
1 //Edilberto Vicente Martínez
2 #include <stdio.h>
3 int main ()
4 {
5     int enteroNumAlumnos = 5;
6     float realCalif = 0.0, realPromedio = 0.0;
7     printf("\tPromedio de calificaciones\n");
8     int indice = 0;
9     while (indice < enteroNumAlumnos)
10    {
11        printf("\nIngrese la calificación del alumno %d\n", indice+1);
12        scanf("%f", &realCalif);
13        realPromedio += realCalif;
14        indice++;
15    }
16    printf("\nEl promedio de las calificaciones ingresadas es: %f\n",
17    realPromedio/enteroNumAlumnos);
18    return 0;
19 }
```

input
8
El promedio de las calificaciones ingresadas es: 7.400000

*En este programa se pide 5 calificaciones al usuario y despues de ello se muestra el promedio de estas calificaciones.

6.-Después de analizar el ejercicio6.c indica para que sirve la cláusula define

La cláusula define nos sirve para definir un valor numérico o texto a una palabra, en el cual si ponemos solamente la palabra en nuestras instrucciones, el programa lo tomará en cuenta como un valor (número) en este caso y no una palabra (caracter).

```
main.c
1 //Edilberto Vicente Martínez
2 #include <stdio.h>
3 #define MAX 5
4 int main ()
5 {
6     int arreglo[MAX], cont;
7     for (cont=0; cont<MAX; cont++)
8     {
9         printf("Ingrese el valor %d del arreglo: ", cont+1);
10        scanf("%i", &arreglo[cont]);
11    }
12    printf("El valor ingresado para cada elemento del arreglo es:\n[" );
13    for (cont=0; cont<MAX; cont++)
14    {
15        printf("%d\t", arreglo[cont]);
16    }
17    printf("]\n");
18    return 0;
19 }
```

input

```
Ingrese el valor 1 del arreglo: 6
Ingrese el valor 2 del arreglo: 9
Ingrese el valor 3 del arreglo: 32
Ingrese el valor 4 del arreglo: 49
Ingrese el valor 5 del arreglo: 26
El valor ingresado para cada elemento del arreglo es:
[6      9      32      49      26      ]
```

*En este programa nos pide 5 valores enteros cualesquiera para poder realizar el arreglo, en donde el valor asignado al arreglo va definido a la palabra MAX.

7.- Actividad: resuelve, codifica el problema y dibuja.

```

INICIO
    ESCRIBIR "Dame la altura"
    LEER n;
    ESCRIBIR "\n";

    SI (n>0) ENTONCES
        PARA (k = 1 ; k<=n-1; k++)
            ESCRIBIR " ";
        FINPARA

        ESCRIBIR "*";
        ESCRIBIR "\n";
    FINSI

    PARA [k=2] HASTA k<=n-1; k++)
        PARA(j = 1; HASTA j <= n-k; j++)
            ESCRIBIR " ";
        FINPARA

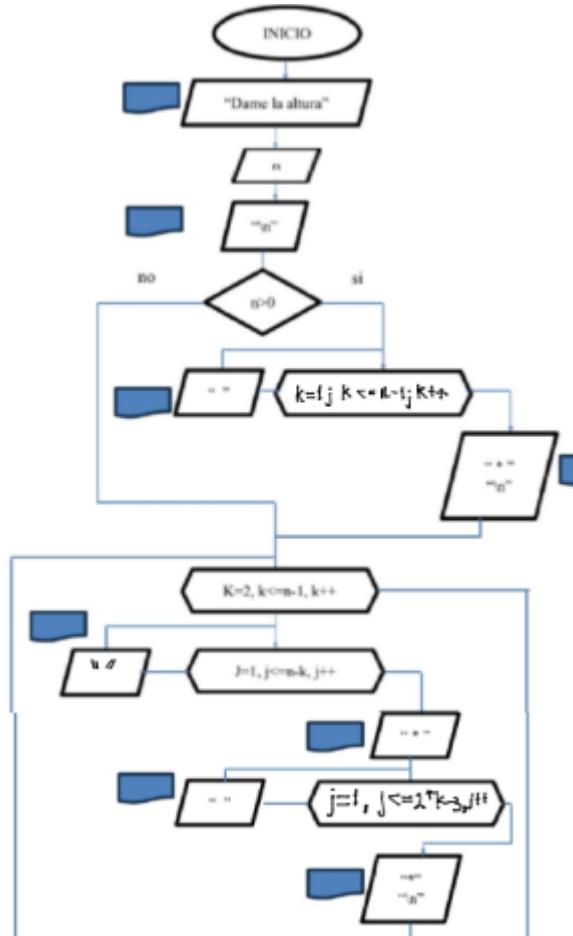
        ESCRIBIR "*";
        ESCRIBIR "\n";
    FINPARA

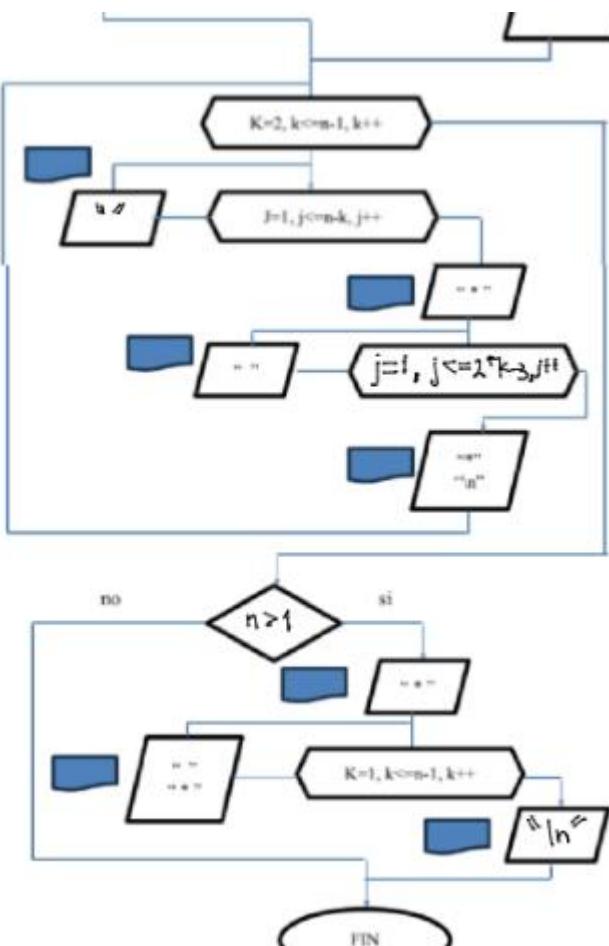
    ESCRIBIR "*";
    ESCRIBIR "\n";

    FINPARA

    SI (n > 1) ENTONCES
        ESCRIBIR "*";
        PARA (k=n-1; K<=n-1; k++)
            ESCRIBIR " ";
        FINPARA
        ESCRIBIR "\n";
    FINSI
FIN

```





```

main.c //Edilberto Vicente Martínez
1 #include <stdio.h>
2 int main() {
3     int n, k, j, c;
4     printf("Dame la altura: ");
5     scanf("%d", &n);
6     printf("\n");
7     if (n > 0) {
8         for (k = 1; k <= n - 1; k++) {
9             for (j = 1; j <= n - k; j++) {
10                printf(" ");
11            }
12            printf("*\n");
13        }
14        for (k = 2; k <= n - 1; k++) {
15            for (j = 1; j <= n - k; j++) {
16                printf(" ");
17            }
18            printf("**");
19            c = 2 * k - 3;
20            for (j = 1; j <= c; j++) {
21                printf(" ");

```

input

```

* **** *

```

```
main.c
14    for (k = 2; k <= n - 1; k++) {
15        for (j = 1; j <= n - k; j++) {
16            printf(" ");
17        }
18        printf("*");
19        c = 2 * k - 3;
20        for (j = 1; j <= c; j++) {
21            printf(" ");
22        }
23        printf("\n");
24    }
25    if (n > 1)
26    {
27        printf("*");
28        for (k = 1; k <= n - 1; k++) {
29            printf("#");
30        }
31        printf("\n");
32    }
33    return 0;
34 }
```

input

Dame la altura: 6

10

```
Dame la altura: 6
      *
     * *
    *   *
   *     *
  *       *
 *         *
* #####
```

Tarea:

1.- Solicitar radio a usuario y calcular el área.

The screenshot shows a terminal window with the following content:

```
main.c
1 //Edilberto Vicente Martínez
2 #include <stdio.h>
3 #define PI 3.14159
4 int main ()
5 {
6     float radio, area;
7     printf("--- 1. Calculo de Area ---\n");
8     printf("Introduce el radio del circulo: ");
9     scanf("%f", &radio);
10    if (radio < 0)
11    {
12        printf("El radio no puede ser negativo.\n");
13    } else
14    {
15        area = PI * radio * radio;
16        printf("El area del circulo con radio %.2f es: %.2f\n", radio, area);
17    }
18 return 0;
19 }
```

input

```
--- 1. Calculo de Area ---
Introduce el radio del circulo: 12
El area del circulo con radio 12.00 es: 452.39
```

2.- Mostrar la numeración del 10 al 1 haciendo uso del ciclo while

The screenshot shows a terminal window with the following content:

```
main.c
1 //Edilberto Vicente Martínez
2 #include <stdio.h>
3 int main()
4 {
5     int i = 10;
6     printf("--- 2. Cuenta Regresiva ---\n");
7     while (i >= 1) {
8         printf("%d\n", i);
9         i--;
10    }
11 return 0;
12 }
```

input

```
--- 2. Cuenta Regresiva ---
10
9
8
7
6
5
```

3.- Mostrar la tabla de multiplicar del número que ingrese el usuario con el ciclo for.

The screenshot shows a terminal window with the following content:

```
main.c
1 //Edilberto Vicente Martínez
2 #include <stdio.h>
3 int main()
4 {
5     int numero, i;
6
7     printf("--- 3. Tabla de Multiplicar ---\n");
8     printf("Introduce un numero para ver su tabla: ");
9     scanf("%d", &numero);
10
11    printf("Tabla del %d:\n", numero);
12    for (i = 1; i <= 10; i++) {
13        printf("%d x %d = %d\n", numero, i, (numero * i));
14    }
15    return 0;
16 }
```

Terminal output:

```
--- 3. Tabla de Multiplicar ---
Introduce un numero para ver su tabla: Tabla del 5:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
```

4.- Genera un menú con los 3 ejercicios anteriores y agrega el ciclo do-while con el fin de que el usuario no pueda salir del programa hasta seleccionar la opción salir en el menú.

The screenshot shows a terminal window with the following content:

```
main.c
1 //Edilberto Vicente Martínez
2 #include <stdio.h>
3 #define PI 3.14159
4 void calcularArea();
5 void contarDiezAlUno();
6 void tablaMultiplicar();
7 int main()
8 {
9     char op;
10
11    do {
12        printf("\n--- MENU PRINCIPAL ---\n");
13        printf("Selecciona un ejercicio:\n");
14        printf("1. Calcular area de un circulo\n");
15        printf("2. Cuenta regresiva de 10 a 1 (con while)\n");
16        printf("3. Tabla de multiplicar (con for)\n");
17        printf("4. Salir\n");
18        printf("Opcion: ");
19
20        scanf(" %c", &op);
21    }
```

Terminal output:

```
--- MENU PRINCIPAL ---
Selecciona un ejercicio:
1. Calcular area de un circulo
2. Cuenta regresiva de 10 a 1 (con while)
3. Tabla de multiplicar (con for)
4. Salir
```

```
main.c
42     return 0;
43 }
44 void calcularArea()
45 {
46     float radio, area;
47
48     printf("\n--- 1. Calculo de Area ---\n");
49     printf("Introduce el radio del circulo: ");
50     scanf("%f", &radio);
51
52     if (radio < 0) {
53         printf("El radio no puede ser negativo.\n");
54     } else {
55         area = PI * radio * radio;
56         printf("El area del circulo con radio %.2f es: %.2f\n", radio, area);
57     }
58 }
59 void contarDiezAUno()
60 {
61     int i = 10;
62 }
```

input
Opcion: 1
--- 1. Calculo de Area ---
Introduce el radio del circulo: 33
El area del circulo con radio 33.00 es: 3421.19

```
main.c
21
22     switch (op) {
23         case '1':
24             calcularArea();
25             break;
26         case '2':
27             contarDiezAUno();
28             break;
29         case '3':
30             tablaMultiplicar();
31             break;
32         case '4':
33             printf("\nSaliendo del programa. ¡Adios!\n");
34             break;
35         default:
36             printf("\n;Opcion no valida! Intenta de nuevo.\n");
37             break;
38     }
39
40 } while (op != '4');
41
42 return 0;
```

input
Opcion: 1
--- 1. Calculo de Area ---
Introduce el radio del circulo: 33
El area del circulo con radio 33.00 es: 3421.19

```
main.c
42     return 0;
43 }
44 void calcularArea()
45 {
46     float radio, area;
47
48     printf("\n--- 1. Calculo de Area ---\n");
49     printf("Introduce el radio del circulo: ");
50     scanf("%f", &radio);
51
52     if (radio < 0) {
53         printf("El radio no puede ser negativo.\n");
54     } else {
55         area = PI * radio * radio;
56         printf("El area del circulo con radio %.2f es: %.2f\n", radio, area);
57     }
58 }
59 void contarDiezAUno()
60 {
61     int i = 10;
62 }
```

input
Opcion: 1
--- 1. Calculo de Area ---
Introduce el radio del circulo: 33
El area del circulo con radio 33.00 es: 3421.19

```
main.c
57 }
58 }
59 void contarDiezAUno()
60 {
61     int i = 10;
62
63     printf("\n--- 2. Cuenta Regresiva (While) ---\n");
64
65     while (i >= 1) {
66         printf("%d\n", i);
67         i--;
68     }
69 }
70 void tablaMultiplicar()
71 {
72     int numero, i;
73
74     printf("\n--- 3. Tabla de Multiplicar (For) ---\n");
75     printf("Introduce un numero para ver su tabla: ");
76     scanf("%d", &numero);
77
78     printf("Tabla del %d:\n", numero);
79 }
```

Opcion: 1

--- 1. Calculo de Area ---
Introduce el radio del circulo: 33
El area del circulo con radio 33.00 es: 3421.19

```
main.c
62
63     printf("\n--- 2. Cuenta Regresiva (While) ---\n");
64
65     while (i >= 1) {
66         printf("%d\n", i);
67         i--;
68     }
69 }
70 void tablaMultiplicar()
71 {
72     int numero, i;
73
74     printf("\n--- 3. Tabla de Multiplicar (For) ---\n");
75     printf("Introduce un numero para ver su tabla: ");
76     scanf("%d", &numero);
77
78     printf("Tabla del %d:\n", numero);
79
80     for (i = 1; i <= 10; i++) {
81         printf("%d x %d = %d\n", numero, i, (numero * i));
82     }
83 }
```

input

2. Cuenta regresiva de 10 a 1 (con while)
3. Tabla de multiplicar (con for)
4. Salir
Opcion: 4

Saliendo del programa. ¡Adios!

5.- Del programa 5.c debe modificarlo para que permita solicitar al usuario el Número de alumnos a evaluar empleando un ciclo while en lugar del ciclo for, deben agregar lo que se pide al programa, no solo ejecutar.

```
main.c F9
1 //Edilberto Vicente Martínez
2 #include <stdio.h>
3 int main ()
4 {
5     int enteroNumAlumnos;
6
7     float realCalif = 0.0, realPromedio = 0.0;
8
9     printf("\tPromedio de calificaciones\n");
10    printf("¿Cuantos alumnos desea evaluar? ");
11    scanf("%d", &enteroNumAlumnos);
12    int indice = 0;
13    while (indice < enteroNumAlumnos)
14    {
15        printf("\nIngrese la calificación del alumno %d\n", indice + 1);
16        scanf("%f", &realCalif);
17        realPromedio += realCalif;
18        indice++;
19    }
20    if (enteroNumAlumnos > 0)
21    {
22        printf("\nEl promedio de las calificaciones ingresadas es: %.2f\n",
23               realPromedio / enteroNumAlumnos);
24    }
25    else
26    {
27        printf("\nNo se ingresaron calificaciones.\n");
28    }
29    return 0;
30 }
```

```
Promedio de calificaciones
¿Cuantos alumnos desea evaluar? 3
Ingrese la calificación del alumno 1
9
```

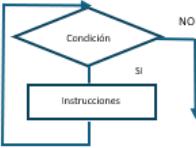
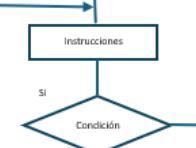
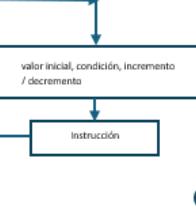
```
main.c F9
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
```

```
Promedio de calificaciones
¿Cuantos alumnos desea evaluar? 
scanf("%d", &enteroNumAlumnos);
int indice = 0;
while (indice < enteroNumAlumnos)
{
    printf("\nIngrese la calificación del alumno %d\n", indice + 1);
    scanf("%f", &realCalif);
    realPromedio += realCalif;
    indice++;
}
if (enteroNumAlumnos > 0)
{
    printf("\nEl promedio de las calificaciones ingresadas es: %.2f\n",
           realPromedio / enteroNumAlumnos);
}
else
{
    printf("\nNo se ingresaron calificaciones.\n");
}
return 0;
```

```
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
```

```
Ingrese la calificación del alumno 3
9
El promedio de las calificaciones ingresadas es: 8.00
```

6.- Completa el cuadro con las estructuras iterativas (MIENTRAS, HACER MIENTRAS Y PARA), agrega solo el apartado en la tarea de la práctica.

Estructura	Estructura de control de flujo	Diagrama de flujo	Pseudocódigo (sintaxis)	Codificación en Lenguaje C (Ejemplo)
Iterativa (repetitiva)	Mientras		MIENTRAS que <expresión lógica> Instrucciones FIN DEL MIENTRAS	while (++cont <=10) printf("%d x%d=%d\n", num, cont, num* cont); return 0; }
	Hacer-mientras		HACER Instrucciones MIENTRAS <expresión lógica>	do { printf ("x=%d\n", x); } while (x < 2);
	Para		PARA valor inicial, condición, incremento / decrecimiento Instrucciones FIN PARA	for (int i=0; i < 2; i++) { printf("i=%d\n", i); }

Conclusión

Esta práctica en lo personal me sirvió para ver el como podemos utilizar las estructuras iterativas ahora en la codificación de nuestros programas, al principio si fue un poco complicado entender el como funcionaban, pero poco a poco me heido adaptando a los distintos tipos de estructuras y a incluir más cosas dentro de ellas, como es el caso de los arreglos. Considero que esta práctica en lo personal ha sido muy buena porqur pude ver claramente las diferencias en cada estructura iterativa.

Bibliografía

*Facultad de Ingeniería. (2025). Manual de prácticas del laboratorio de Fundamentos de Programación. Laboratorio de computación. Salas A y B. Universidad Nacional Autónoma de México. pp. 36-51. Recuperado el 28 de octubre de 2025 de <http://lcp02.fi-b.unam.mx/>

*El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA,

Pearson Educación 1991.