	<b>Carátula para entrega de prácticas</b>	
Facultad de Ingeniería	Laboratorio de docencia	

# Laboratorios de computación salas A y B

*Profesor:* Karina García Morales

*Asignatura:* Fundamentos de programación

*Grupo:* 22

*No. de práctica(s):* Práctica No.8

*Integrante(s):* Freddy Beckham Cedillo Arias

*No. de lista o brigada:* No.11

*Semestre:* 1er Semestre

*Fecha de entrega:* 8 de octubre del 2024

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

# Práctica 8

## Estructuras de repetición

### Objetivo:

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

### Desarrollo :

- Tres estructuras de control iterativas son:

*while* =

La estructura repetitiva (o iterativa) *while* primero valida la expresión lógica y si ésta se cumple (es verdadera) procede a ejecutar el bloque de instrucciones de la estructura, el cual está delimitado por las llaves {} y regresa a validar la condición nuevamente, esto lo realiza mientras la condición sea verdadera. Cuando la condición no se cumple se continúa el flujo normal del programa sin ejecutar el bloque de la estructura, es decir, el bloque se puede ejecutar de cero a un determinado número de veces.

*do-while* =

*do-while* es una 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 un determinado número de veces.

*for* =

El lenguaje C posee la estructura de repetición *for* la cual permite realizar repeticiones cuando generalmente el control de la repetición está definido sobre una variable contador.

- Diferencia entre *while* y *do-while*:

En la estructura *do while*, la condición se encuentra debajo del bloque a repetir, a diferencia de *while* o *for*, donde se encuentra en la parte superior.

## Tarea :

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

```
5     float radio, area;  
6  
7     printf("Ingrese el radio del círculo: ");  
8     scanf("%f", &radio);  
9  
10    area = PI * radio * radio;  
11  
12    printf("El área del círculo es: %.2f\n", area);  
13  
14    return 0;  
15 }
```

```
1 #include <stdio.h>  
2 #define PI 3.1416  
3  
4 int main() {  
5     float radio, area;  
6  
7     printf("Ingrese el radio del círculo: ");  
8     scanf("%f", &radio);  
9  
10    area = PI * radio * radio;  
11 }
```

```
Ingrese el radio del círculo: El área de  
[Execution complete with exit code 0]
```

```
el círculo: El área del círculo es: 0.00  
e with exit code 0]
```

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

```

1 #include <stdio.h>
2
3 int main() {
4     int num = 10;
5
6     while (num > 0) {
7         printf("%d\n", num);
8         num--;
9     }
10
11     return 0;
12 }

```

```

10
9
8
7
6
5
4
3
2
1

[Execution complete with exit code 0]

```

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

```

1 #include <stdio.h>
2
3 int main() {
4     int numero, i;
5
6     printf("Ingrese un número para mostrar su tabla de multiplicar: ");
7     scanf("%d", &numero);
8
9     for (i = 1; i <= 10; i++) {
10        printf("%d x %d = %d\n", numero, i, numero * i);
11    }
12
13    return 0;
14 }

```

```

1
2
3
4
5
6 a mostrar su tabla de multiplicar: ");
7
8
9
10 numero, i, numero * i);
11
12
13
14

```

```

Ingrese un número para mostrar su tabla
0 x 2 = 0
0 x 3 = 0
0 x 4 = 0
0 x 5 = 0
0 x 6 = 0
0 x 7 = 0
0 x 8 = 0
0 x 9 = 0
0 x 10 = 0

[Execution complete with exit code 0]

```

```

strar su tabla de multiplicar: 0 x 1 = 0

exit code 0]

```

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 en el menú

```

1 #include <stdio.h>
2 #define PI 3.1416
3
4 void calcularArea() {
5     float radio, area;
6     printf("Ingrese el radio del círculo: ");
7     scanf("%f", &radio);
8     area = PI * radio * radio;
9     printf("El área del círculo es: %.2f\n", area);
10 }
11

```

```

12 void cuentaRegresiva() {
13     int num = 10;
14     while (num > 0) {
15         printf("%d\n", num);
16         num--;
17     }
18 }
19
20 void tablaMultiplicar() {
21     int numero, i;
22     printf("Ingrese un número para mostrar su tabla de multiplicar: ");
23

```

```

24     printf("Ingrese un número para mostrar su tabla de multiplicar: ");
25     scanf("%d", &numero);
26     for (i = 1; i <= 10; i++) {
27         printf("%d x %d = %d\n", numero, i, numero * i);
28     }
29 }
30
31 int main() {
32     int opcion;
33     do {
34         printf("\nMenú de opciones:\n");
35

```

```

36         printf("\nMenú de opciones:\n");
37         printf("1. Calcular el área de un círculo\n");
38         printf("2. Contar del 10 al 1\n");
39         printf("3. Mostrar tabla de multiplicar\n");
40         printf("4. Salir\n");
41         printf("Seleccione una opción: ");
42         scanf("%d", &opcion);
43
44         switch (opcion) {
45             case 1:
46                 calcularArea();
47

```

```

41         case 1:
42             calcularArea();
43             break;
44         case 2:
45             cuentaRegresiva();
46             break;
47         case 3:
48             tablaMultiplicar();
49             break;
50         case 4:
51             printf("Saliendo del programa...\n");

```

```

47         break;
50         case 4:
51             printf("Saliendo del programa...\n");
52             break;
53         default:
54             printf("Opción no válida, por favor intente de nuevo.\n");
55     }
56 } while (opcion != 4);
57
58 return 0;
59 }

```

5.- Del ejercicio de la calculadora, agregar condición a la división, si hay división entre cero pedir otro valor al usuario hasta que coloque un valor diferente y se pueda realizar la división.

```

#include <stdio.h>

int main() {
    int op, uno, dos;

    do {
        printf("----Calculadora---\n");
        printf("1) Sumar\n");
        printf("2) Restar\n");
        printf("3) Multiplicar\n");
        printf("4) Dividir\n");
        printf("5) Salir\n");
        scanf("%d", &op);

        switch (op) {
            case 1:
                printf("Introduzca los números a sumar separados por comas: ");
                scanf("%d %d", &uno, &dos);
                printf("%d + %d = %d\n", uno, dos, (uno + dos));
                break;

            case 2:
                printf("Introduzca los números a restar separados por comas: ");
                scanf("%d %d", &uno, &dos);
                printf("%d - %d = %d\n", uno, dos, (uno - dos));
                break;

            case 3:
                printf("Introduzca los números a multiplicar separados por comas: ");
                scanf("%d %d", &uno, &dos);
                printf("%d * %d = %d\n", uno, dos, (uno * dos));
                break;

            case 4:
                printf("Introduzca los números a dividir separados por comas: ");
                scanf("%d %d", &uno, &dos);
                while (dos == 0) {
                    printf("Error: No se puede dividir entre cero. Ingrese otro divisor:");
                    scanf("%d", &dos);
                }
                printf("%d / %d = %.2f\n", uno, dos, (double)uno / dos);
                break;

            case 5:
                printf("Salir\n");
                break;

            default:
                printf("Opción inválida.\n");
        }
    } while (op != 5);

    return 0;
}

```

```

---Calculadora---
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
5
Salir

...Program finished with exit code 0
Press ENTER to exit console.

```

6.- Modificar el programa para que permita solicitar al usuario el Número de alumnos a evaluar empleando un ciclo while en lugar del ciclo for

### Programa5.c

```

#include <stdio.h>
int main ()
{
    int enteroNumAlumnos = 5;
    float realCalif = 0.0, realPromedio = 0.0;
    printf("\tPromedio de calificaciones\n");
    for (int indice = 0 ; indice < enteroNumAlumnos ; indice++)
    {
        printf("\nIngrese la calificación del alumn %d\n", indice+1);
        scanf("%f",&realCalif);
        realPromedio += realCalif;
    }
    printf("\nEl promedio de las calificaciones ingresadas es: %f\n",
        realPromedio/enteroNumAlumnos);
    return 0;
}

```

### Ciclo while

```

1 #include <stdio.h>
2
3 int main() {
4     int enteroNumAlumnos, indice = 0;
5     float realCalif = 0.0, realPromedio = 0.0;
6     printf("Ingrese el número de alumnos a evaluar: ");
7     scanf("%d", &enteroNumAlumnos);
8
9     while (indice < enteroNumAlumnos) {
10         printf("\nIngrese la calificación del alumno %d: ", indice + 1);
11         scanf("%f", &realCalif);

```

```

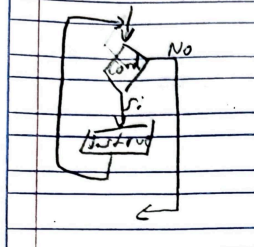
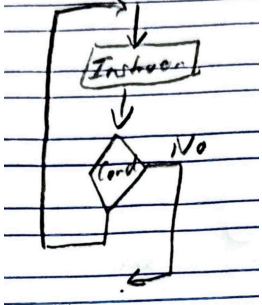
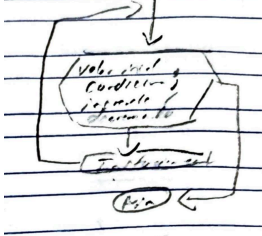
12     }
13     printf("\nEl promedio de las calificaciones ingresadas es: %.2f\n", realPromedio / enteroNumAlumnos);
14     return 0;
15 }

```

Ingrese el número de alumnos a evaluar:  
El promedio de las calificaciones ingresadas es:  
[Execution complete with exit code 0]

mero de alumnos a evaluar:  
e las calificaciones ingresadas es: -nan  
mplete with exit code 0]

## 7.- Completa el cuadro con las estructuras iterativas (MIENTRAS, HACER MIENTRAS Y PARA)

Estructura	Diagrama	Pseudocódigo	Codificación
Iterativa Mientras		MIENTRAS que <expresión lógica> Instrucciones FIN DEL MIENTRAS	While(expresión) { Secuencia de instrucciones; }
Iterativa Hacer Mientras		Hacer <instrucciones> Mientras <condición>	do { /*Bloque de código que se ejecuta por lo menos una vez y se repite mientras la expresión lógica sea verdadera.*/ } while (expresión_ lógica);
Iterativa Para		Para <variable> desde <valor inicial> hasta <valor final> hacer <instrucciones> Fin Para	for (inicialización; expresión lógica ; operaciones por iteración) { /*Bloque de código a ejecutar*/ }



## **Conclusión:**

En conclusión estos nuevos códigos nos son de gran ayuda gracias a que nos ayudan a tener mayor control y flexibilidad en los bucles; por ejemplo el ciclo for se utiliza cuando conocemos cuántas veces iterarás, el while es útil cuando la repetición depende de una condición que debe cumplirse antes de cada iteración, y el do-while nos ayuda que el bloque de código se ejecute al menos una vez.

## **Bibliografía :**

Moisset, D. (s. f.). Estructura repetitiva do while. Recuperado de :

<https://www.tutorialesprogramacionya.com/cya/detalleconcepto.php?punto=11&codigo=11&inicio=0>

Laboratorio Salas A y B. (s. f.). Manual de práctica de fundamentos de programación;

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