



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

---

*Profesor: Alejandro Esteban Pimentel Alarcon*

*Asignatura: Fundamentos de programación*

*Grupo: 3*

*No de Práctica(s): 9*

*Integrante(s): Rivera Sosa Arlethe*

*No. de Equipo de cómputo empleado:*

*No. de Lista o Brigada: 317083033*

*Semestre: 2020-1*

*Fecha de entrega: 13/octubre/19*

*Observaciones:*    Muy bien

**CALIFICACIÓN: 10**

## Programas en C

La estructura de repetición o bucle hace posible la ejecución repetida de una o más instrucciones. Las estructuras de repetición nos permiten ejecutar varias veces unas mismas líneas de código. Estas estructuras describen procesos que se repiten varias veces en la solución del problema. El conjunto de acciones que se repiten conforma el cuerpo del bucle y cada ejecución del cuerpo del bucle se denomina iteración.

La instrucción while es una de las alternativas para la repetición de procesos en programación. Guarda cierta similitud con la instrucción for, hasta el punto de que en algunas ocasiones podrá optarse por el uso de una u otra de forma indistinta. While es una palabra clave en C que admite varios usos. Las instrucciones while son anidables dentro de sí mismas o dentro de otras estructuras. Es importante verificar que los bucles diseñados con esta instrucción dispongan de una condición.

El bucle do – while se puede leer como «haz – mientras», esta estructura se encarga de repetir de forma cíclica un conjunto de instrucciones que se encuentren dentro del bucle, esta repetición se realizará hasta que se cumpla la condición de parada que definamos y que es evaluada con la palabra reservada while. El ciclo do-while nos permite añadir cierta ventaja adicional y esta consiste que nos da la posibilidad de ejecutar primero el bloque de instrucciones antes de evaluar la condición necesaria, de este modo los ciclos do-while, son más efectivos para algunas situaciones específicas.

FOR es una estructura de repetición empleada en la programación de algoritmos para repetir un código una o más veces dependiendo de un contador. For permite controlar el flujo de ejecución de sentencias de programación permitiendo repetir un código. La estructura For posee un contador explícito, esto permite al cuerpo del For (el código que es ejecutado repetidas veces) saber en qué repetición está. Generalmente la estructura For se emplea cuando se sabe de antemano el número de iteraciones necesarias antes de entrar a las repeticiones. Se puede construir un For empleando un While con un contador y agregando la condición de "corte" asociada al contador en el While.

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

Actividades:

- Hacer un programa que pida un número y muestre su tabla de multiplicar (hasta el 10).

```

1  # include <stdio.h>
2  int main(){
3      int tabla;
4      int i=1;
5      printf("\n ingresa un número");
6      scanf("%d", &tabla);
7      puts("");
8      printf("tabla del %d\n", tabla);
9      while(i<= 10){
10         printf("%d*%d= %d\n", tabla, i, tabla*i);
11         i=i+1;}
12         return 0;
13     }

```

```

Familia@DESKTOP-NE4QAFJ ~
$ gcc tabla.c -o tabla

Familia@DESKTOP-NE4QAFJ ~
$ ./tabla

ingresa un número 5

tabla del 5
5*1= 5
5*2= 10
5*3= 15
5*4= 20
5*5= 25
5*6= 30
5*7= 35
5*8= 40
5*9= 45
5*10= 50

```

En este programa ocupamos while repitiendo un ciclo para que nos sacara la tabla de multiplicar de cualquier número hasta llegar al 10.

- Hacer un programa que pida y lea 10 números y muestre su suma y su promedio.

```

1  #include <stdio.h>
2  #define prom 10
3  int main(){
4      int num;
5      float promedio;
6      int z;
7      int i;
8      do{
9          printf("escribe 10 numeros\n");
10         int i=1;
11         int z=0;
12     }while(i<10);
13         scanf("%i", &num);
14         z= num+z;
15         i++;
16     promedio:(double) z/prom;
17     printf("la suma es de: %i\n", z);
18     printf("el promedio: %.2f\n", promedio);
19     return 0;
20 }

```

```

Familia@DESKTOP-NE4QAF3 ~
$ gcc promedio.c -o prom

Familia@DESKTOP-NE4QAF3 ~
$ ./prom
escribe 10 numeros
5 9 3 4 8 3 6 4 4 2
la suma es de: 48
el promedio: 4.80

```

Utilizamos el do-while para que se repitiera nuestra acción. Definimos 10 para que se secase el promedio, ya que siempre se divide entre 10 al recibir 10 números.

- Hacer un programa que pida un número e indique si es primo o no.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(){
4      int num1,i,a;
5      a=0;
6      printf("intruduce un número:");
7      scanf("%d", &num1);
8      for(i=1; i<=num1; i++){
9          if(num1%i==0){
10             a++;
11         }
12     }
13     if(a==2){
14         printf("número primo");
15     }else{
16
17         printf("el número no es primo");
18     }
19
20     return 0;
21 }

```

```

Familia@DESKTOP-NE4QAFJ ~
$ gcc primo.c -o primo

Familia@DESKTOP-NE4QAFJ ~
$ ./primo
intruduce un número:6
el número no es primo

```

Ocupamos for para saber si nuestro número se podía dividir en otro número.

Conclusión: existen diferentes repetidores y cada uno tiene una función diferente, a pesar de que algunos pueden ocuparse de la misma manera o ser remplazados por otros. Es importante saber que tipo de programa requerimos para así saber que tipo de repetidor vamos a usar.