	Carátula para entrega de prácticas	
Facultad de Ingeniería	Laboratorio de docencia	

Laboratorios de computación salas A y B

Ing. Karina García Morales
Profesor: _____
 Fundamentos de Programación
Asignatura: _____
 1121
Grupo: _____
 9
No de Práctica(s): _____
 Jacinto Rodríguez Moisés Rodrigo
Integrante(s): _____

*No. de Equipo de cómputo
empleado* 24

 2019-1
Semestre: _____
 23/10/2018
Fecha de entrega: _____

Obervaciones: _____

CALIFICACIÓN:

Práctica 9 Estructuras de repetición

Objetivo

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

Desarrollo

Pseudocódigo

Ciclo PARA (for)

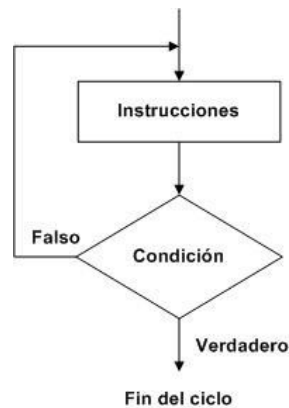
for(expresión lógica) expresión incrementada.

instrucción 1

instrucción 2

fin para

Diagrama de flujo



Ciclo MIENTRAS (while)

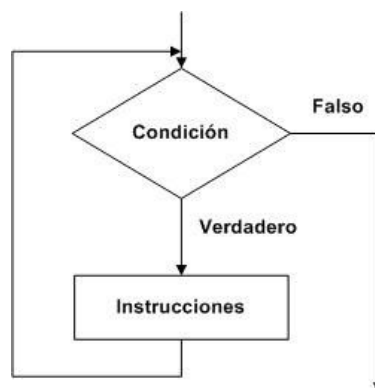
Pseudocódigo

while(condiciones);

Instrucciones

El ciclo while no tiene llaves porque no hay otra instrucción a realizar, cuando hay más de una instrucción a realizar si es necesario el uso de las llaves.

Diagrama de flujo

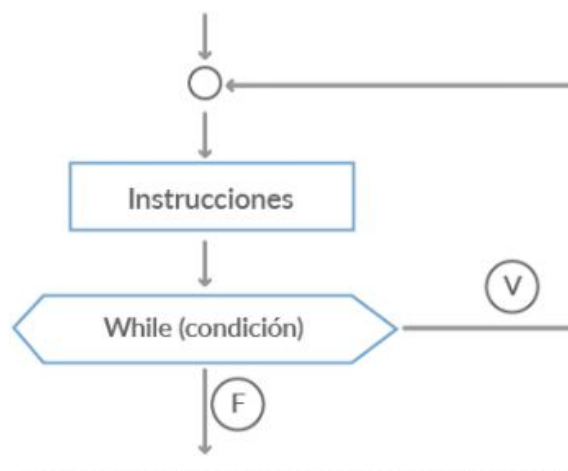


Ciclo HACER MIENTRAS (do-while)

Pseudocódigo

```
do {  
  (sentencia);  
} while (condición);  
do {  
  sentencia 1;  
  sentencia 2;  
} while (condición);
```

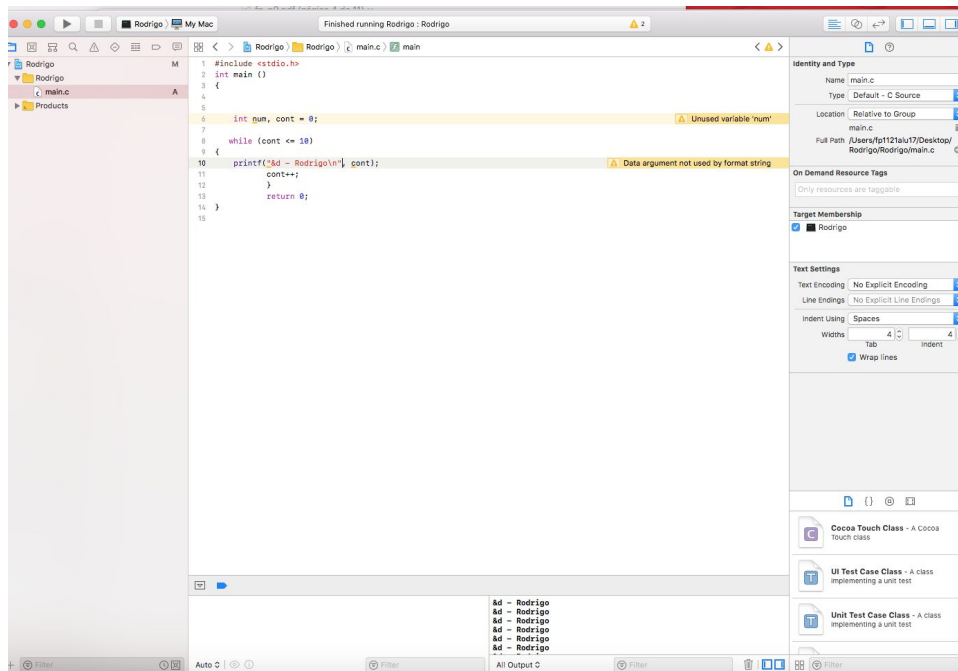
Diagrama de flujo



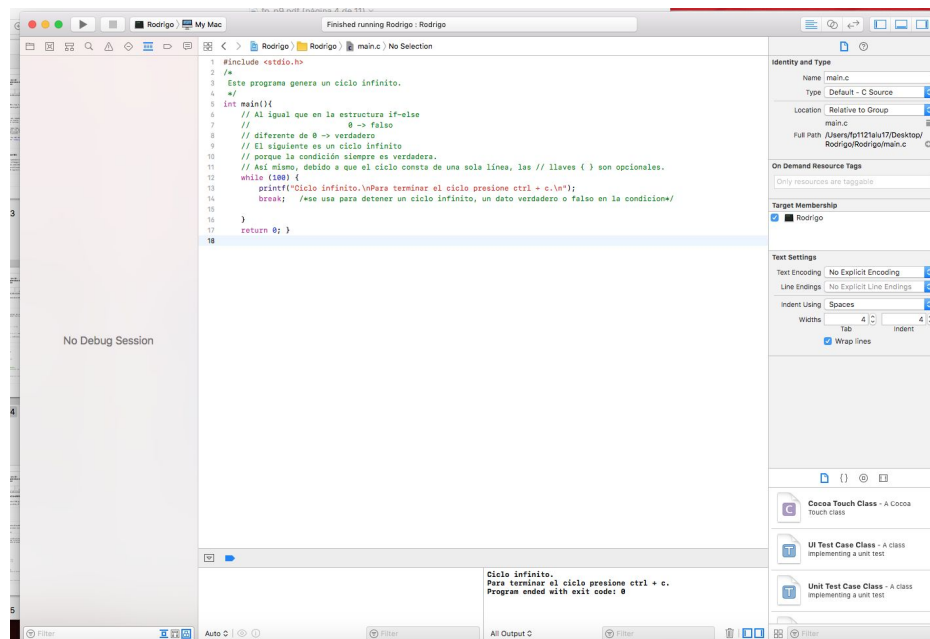
Actividad

Hacer un programa que imprima 10 veces mi nombre y que muestre el número de veces que fue impreso.

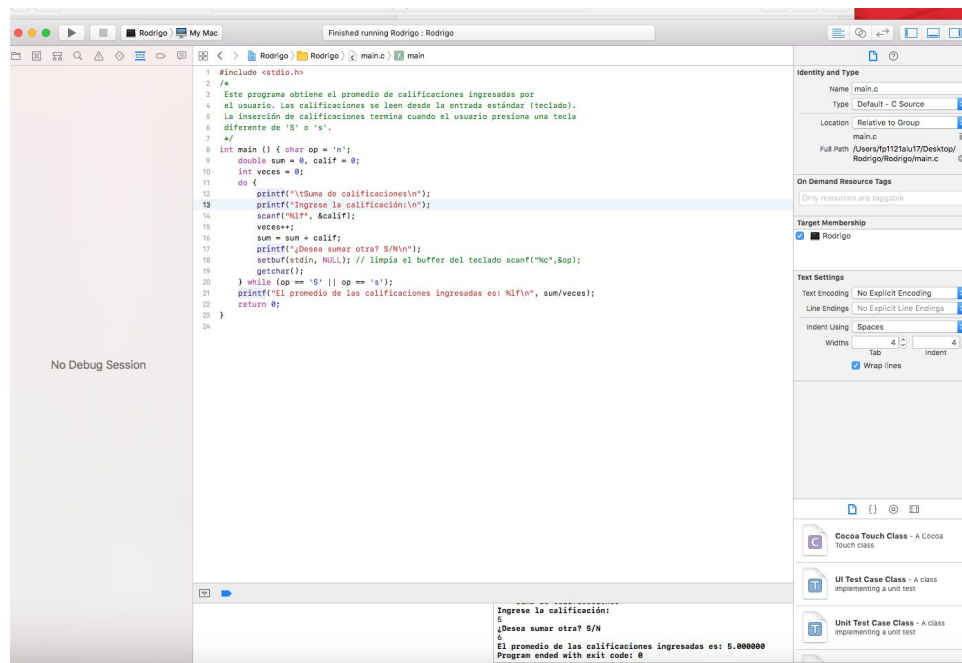
```
#include <stdio.h>  
int main ()  
{  
  int num, cont = 0;  
  while (cont <= 10)  
  {  
    printf("&d - Rodrigo. \n", cont);  
    cont++;  
  }  
  return 0;  
}
```



Uso de la palabra reservada “break”



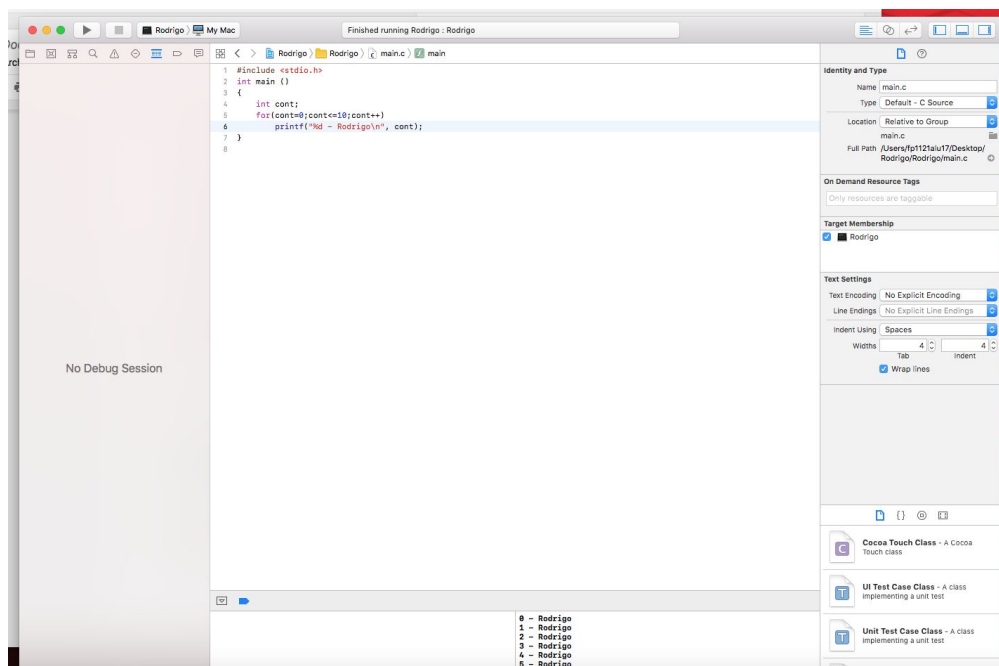
El ciclo do-while es el único que hace uso de “,” y tiene llaves porque las instrucciones serán más de una.



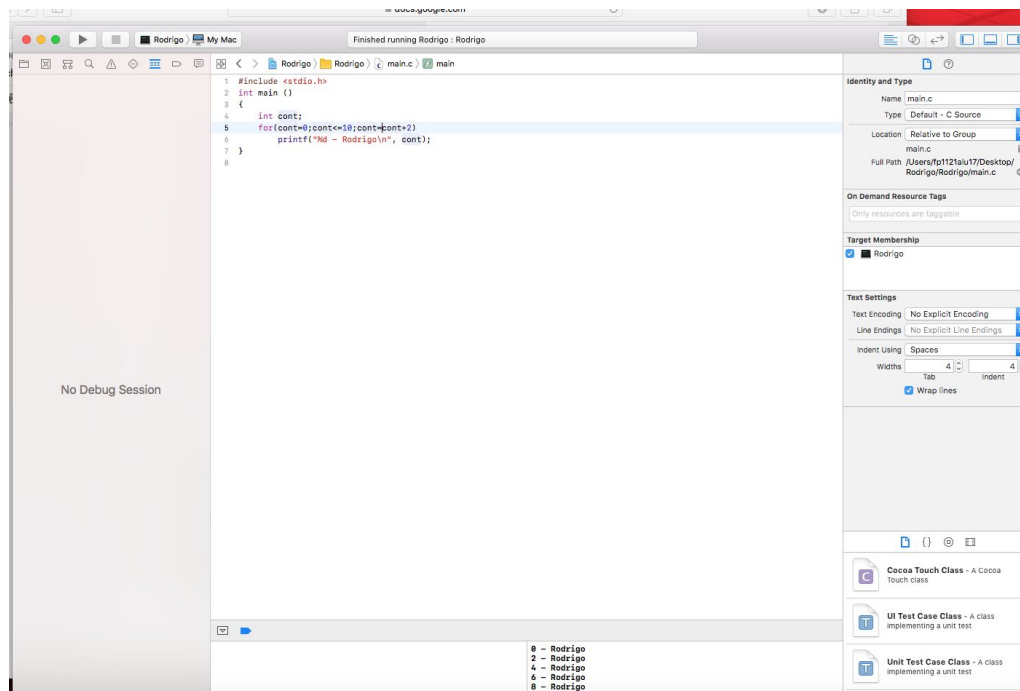
En el ciclo for se insertan tres datos; usa llaves porque hace más de una indicación, si solo hiciera una no necesita el uso de las llaves.

En la declaración de variable, pide los datos, incremento o decremento.

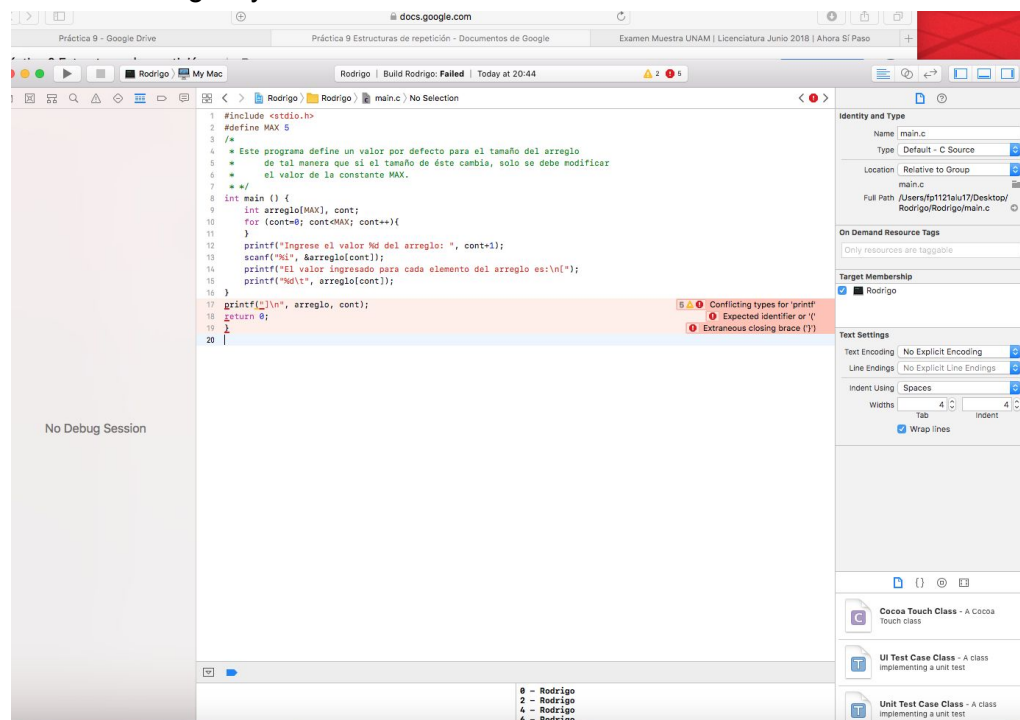
El contador indica que se imprima 10 veces el mensaje, en este caso, mi nombre.



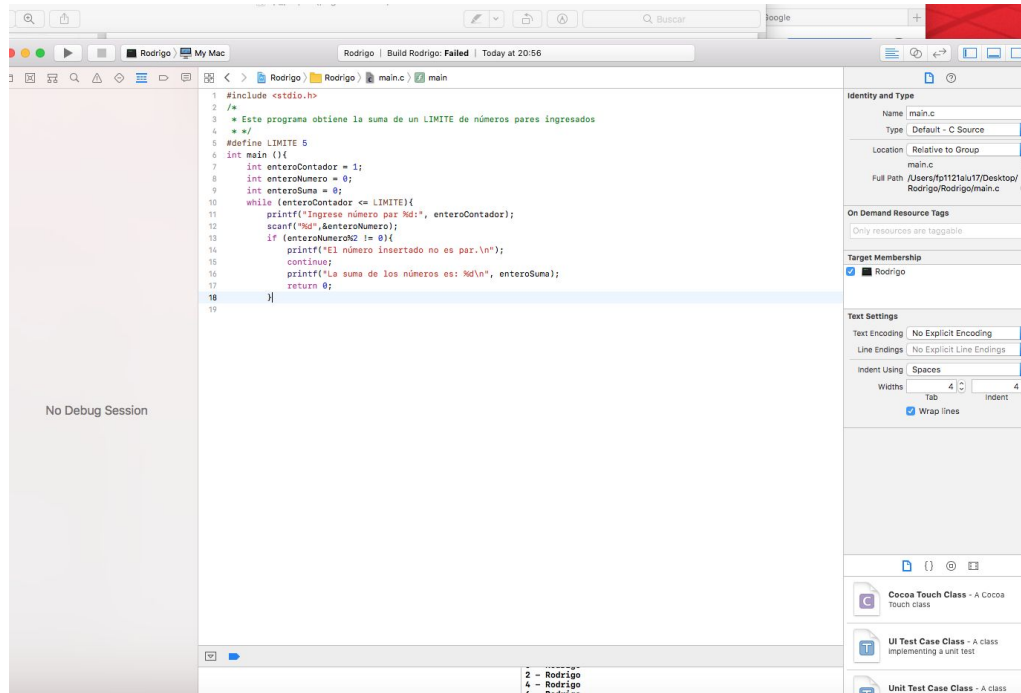
La modificación del contador provoca que la impresión se manifieste en los números indicados por el contador.



Hace uso de los arreglos y el contador



“continue” sirve para dar seguimiento a los 5 datos siempre y cuando la condición se cumpla.



TAREA

Realizar un ciclo con menú.

```
#include<stdio.h>
```

```
int main ()
```

```
{
```

```
int opcion;
```

```
printf ("1) opción 1\n");
```

```
printf ("2) opción 2\n");
```

```
printf ("3) opción 3\n");
```

```
printf ("4) Salir\n");
```

```
switch(opcion) {
```

```
case 1 ("Usted seleccionó %d\n", opcion);
```

```
break;
```

```
case 2 ("Usted seleccionó %d\n", opcion);
```

```
break;
```

```
case 3 ("Usted seleccionó %d\n", opcion);
```

```
break;
```

```
case 4
```

```
break;
```

```
default
```

```
printf ("Opcion incorrecta.\n");
```

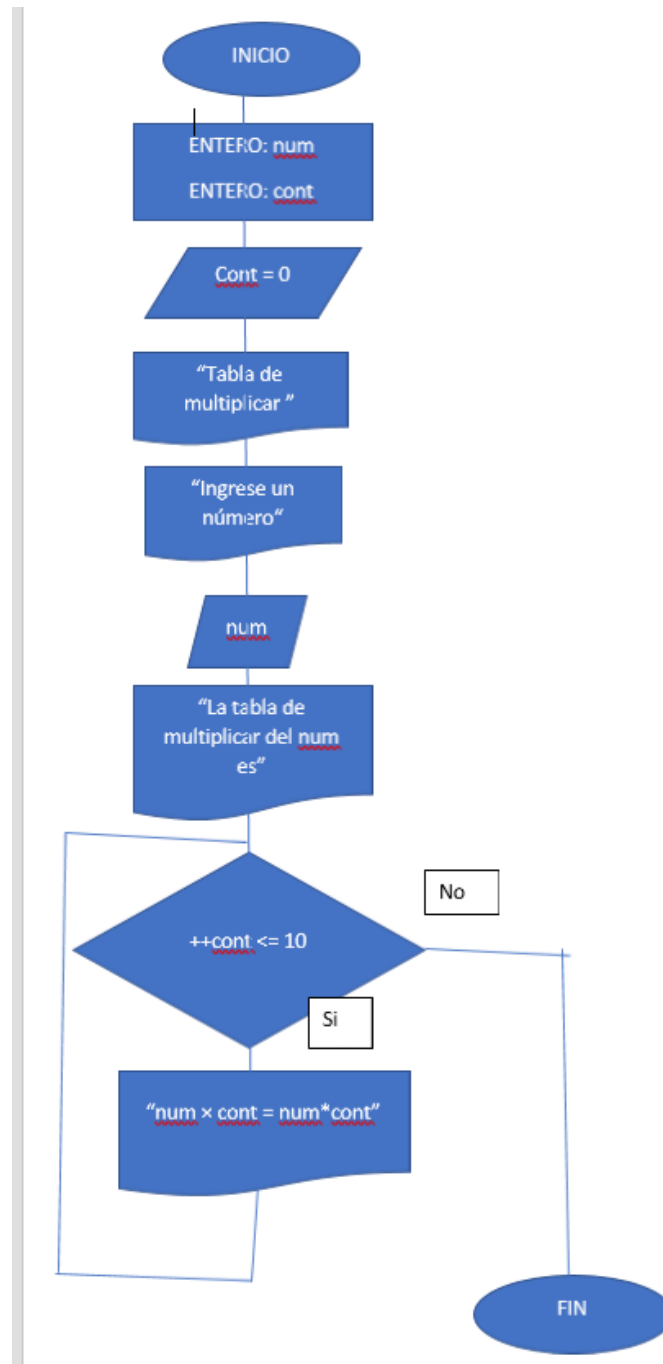
```

break;
}
}while(opcion !=4);
return 0;
}

```

Ejercicios propuestos

1.- Dibujar el diagrama de flujo del programa para obtener la tabla de multiplicar con ciclo While.



2.- Cambiar el ejercicio de la calculadora, página 5 y agregar la pregunta de la opción si desea calcular o no (do{}while (op == 'S' || op == 's'));

```
#include <stdio.h>
```

```
/* Este programa genera una calculadora básica. */
```

```
int main () {
```

```
    int op, uno, dos;
```

```
    do {
```

```
        printf(" --- Calculadora ---\n");
```

```
        printf("\n¿Qué desea hacer\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("\tSumar\n");
```

```
            printf("Introduzca los números a sumar separados por comas\n");
```

```
            scanf("%d, %d",&uno, &dos);
```

```
            printf("%d + %d = %d\n", uno, dos, (uno + dos));
```

```
            break;
```

```
        case 2:
```

```
            printf("\tRestar\n");
```

```
            printf("Introduzca los números a restar separados por comas\n");
```

```
            scanf("%d, %d",&uno, &dos);
```

```
            printf("%d - %d = %d\n", uno, dos, (uno - dos));
```

```
            break;
```

```
        case 3:
```

```
            printf("\tMultiplicar\n");
```

```
            printf("Introduzca los números a multiplicar separados por comas\n");
```

```
            scanf("%d, %d",&uno, &dos);
```

```
            printf("%d * %d = %d\n", uno, dos, (uno * dos));
```

```
            break;
```

```
        case 4:
```

```
            printf("\tDividir\n");
```

```
            printf("Introduzca los números a dividir separados por comas\n");
```

```
            scanf("%d, %d",&uno, &dos);
```

```
            printf("%d / %d = %.2f\n", uno, dos, ((double)uno / dos));
```

```
            break;
```

```
        case 5:
```

```
            printf("\tSalir\n");
```

```
            break;
```

```
        default:
```

```
printf("\tOpción inválida.\n");  
}  
} while (op != 5);  
return 0;  
}
```

2.- Explica qué sucede con el ciclo while al colocarle un valor positivo, negativo y cero y porque razón se obtiene ese resultado.

Cuando se le coloca un valor positivo verifica que la condición se cumpla y la realiza el número de veces indicadas por el problema.

Al momento de colocar un valor negativo este no se ejecuta y lo marca como un error porque evidentemente se realiza una iteración positiva

Cuando se coloca un valor cero, sólo la condición acepta el valor cero para su ejecución en el programa.

3.- ¿Qué sucede con el ejercicio que ejecuta el break, si se omite y qué sucede si se coloca?

Si se omite break se ejecuta la instrucción de una manera infinita al mostrarse el programa al usuario.

Al colocar el break es para detener la instrucción dada de manera correcta en como lo desea manifestar el programa

4.- ¿Qué sucede con el ejercicio que ejecuta continue, si se omite y qué sucede si se coloca?

Cuando se coloca se inicia la siguiente iteración del ciclo de repetición que la contiene.

Cuando se omite no inicia la iteración del otro ciclo.

Conclusión

A lo largo de esta práctica se vieron los ciclos más elementales para llevar a cabo diferentes tareas que pueden hacer uso de estas estructuras para optimizar procesos de larga duración en diferentes áreas de la ingeniería.

Estas estructuras son de vital importancia, por ende la comprensión y escritura de las mismas dentro de los programas son elementales, es importante destacar que el hecho de practicar este tipo de ejercicios con claras y sencillas explicaciones sirven de bastante ayuda para su comprensión y asimilación para ser empleados en distintos trabajos de diferente índole.