|  |  |  |
| --- | --- | --- |
| escudofi_color_m2008_jpg | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorios de docencia |

Laboratorio de computación . Salas A Y B

|  |  |
| --- | --- |
| Profesor: | Claudia Rodríguez Espinoza |
| Asignatura: | Fundamentos de programación |
| Grupo: | 1104 |
| No de Práctica(s): | N° 9 |
| Integrante(s): | Romero Bernal Rocío Fabiola |
|  |  |
|  |  |
|  |  |
| No. de Equipo de cómputo empleado: | N° 42 |
| Semestre: | 2019-1 |
| Fecha de entrega: | 15/10/18 |
| Observaciones: |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

PRACTICA°9: “Estructuras de repeticion”

OBJETIVOS:Elaborar programas en C para la resolucion de problemas basicos que incluyan las estructuras de repeticion y la directiva define.

Algunas veces empleamos los procesos repetitivos dentro de los programas para obtener en pantalla por ejemplo una serie de valores que pertencen a un intervalo (definido por el programador) mediante una misma variable, aunque tambien es muy util al momento de obtener un resultado que requiere la realizacion de una serie de subprocesos aplicados a un mismo valor inicial, evitando que caigamos en el mal habito de escribir la misma operacion el numero de veces que debemos repetir ese paso a seguir. Por supuesto, sus aplicaciones no se detienen aquí, tanto asi que incluso en las cadenas de caracteres y los arreglos son indispensables para recorrer por cada uno de los valores que pertenecen a dicho conjunto como un contador(en el caso de for) que va seleccionando de uno a uno (o como se le haya indicado) los valores que se encuentren para realizar el proceso correspondiente.

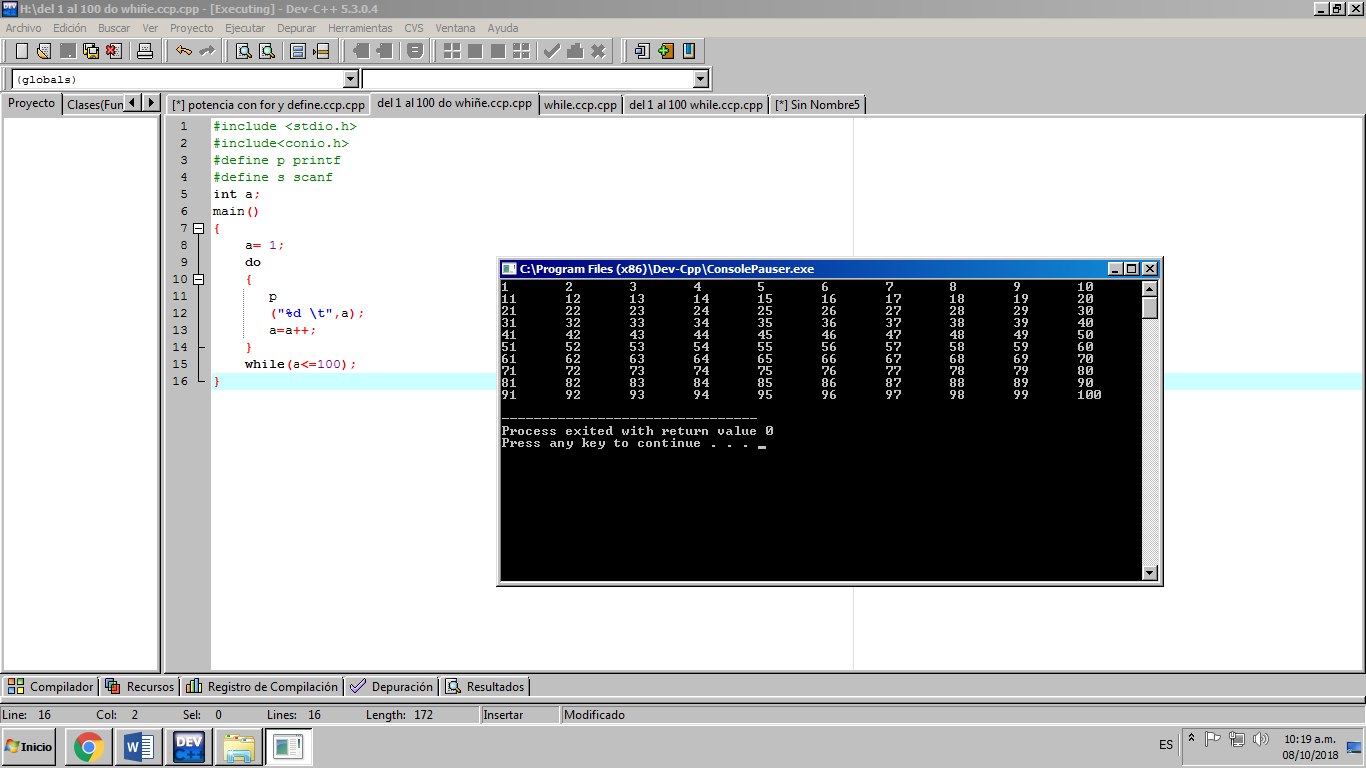
No obstante debemos aprender las diferencias que existen entre las llamadas estructuras de control repetitivas, para poder utilizarlas correctamente, y en el mejor de los casos evitar ciclos infinitos, debido a un mal uso de las mismas.

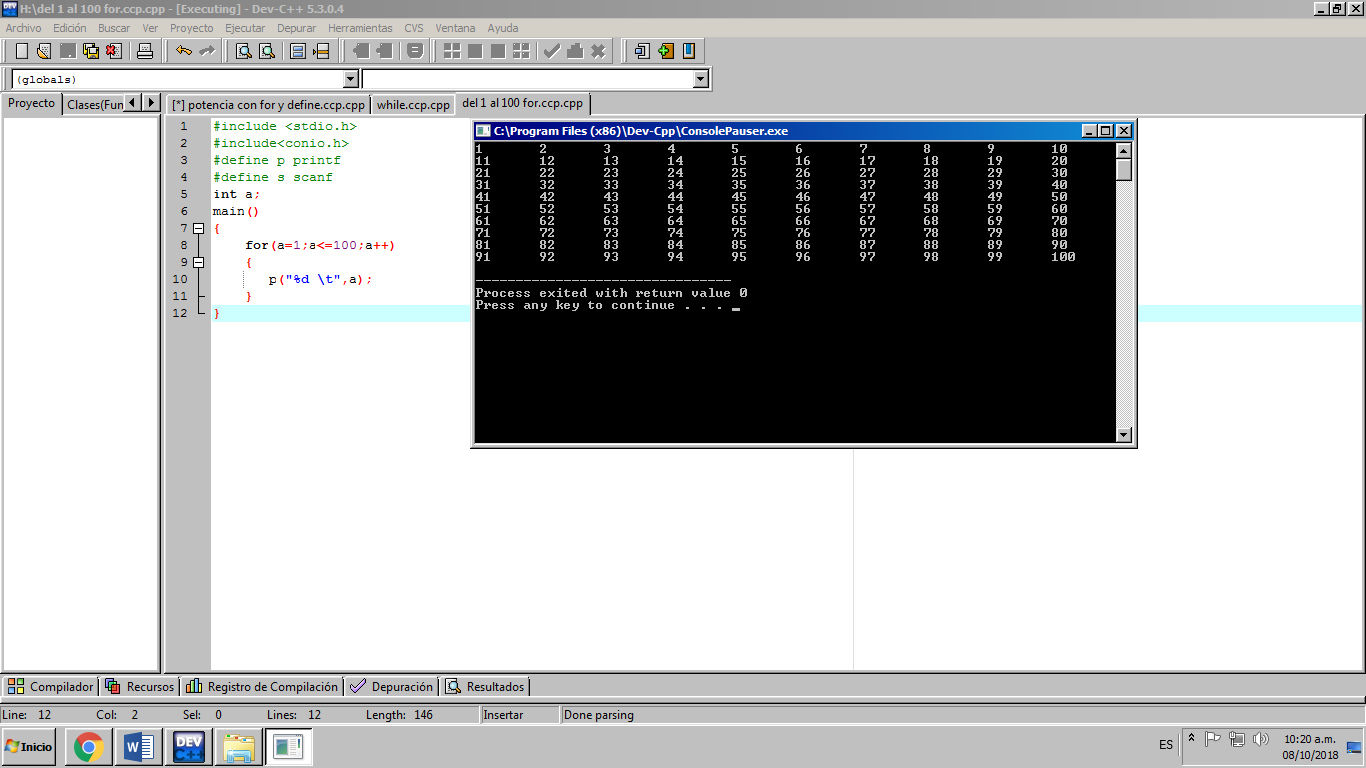
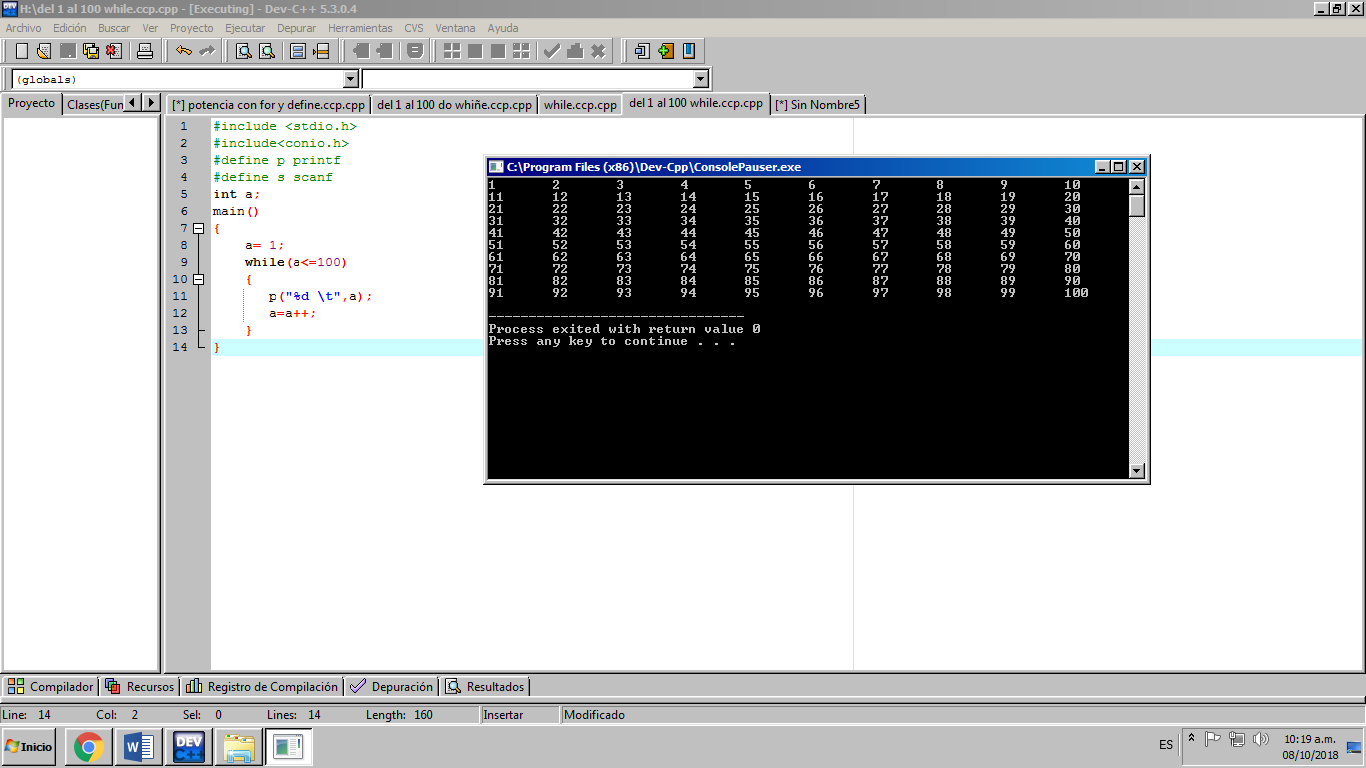
Cuando nosotros nececitemos realizar un proceso el numero de veces que se cumpla la condicion establecida, utilizaremos para ello, la estructura while la cual primero valida la expresion indicada para despues, en caso de que esta se cumpla, realizar el bloque de instrucciones establecido(llegando a repetir el mismo proceso de cero a ene veces) al contrario de su equivalente la estructura do while, que se caracteriza por realizar primero el bloque de instrucciones y posteriomente evaluar la expresion(ejecutandose de una a ene veces).

Y finalmente pero no menos importante se hablara un poco de la estructura de control for, que tal como su nombre lo indica, implica un mayor control para el numero de repeticiones establecidas permitiendo al usuario introducir un valor incial, un valor final y un incremento o decremento. Esta, pricipalmente es muy util cuando se quiere anidar un proceso repetitivo controlado con una cantidad de variables determinadas, sintetizando asi la instrucción que se le da al programa.

Ahora que ya se conoce mas objetivamente el funcionamiento de cada una de las estructuras de control, vamos a emplearlas en los programas que se presentan a continuacion:

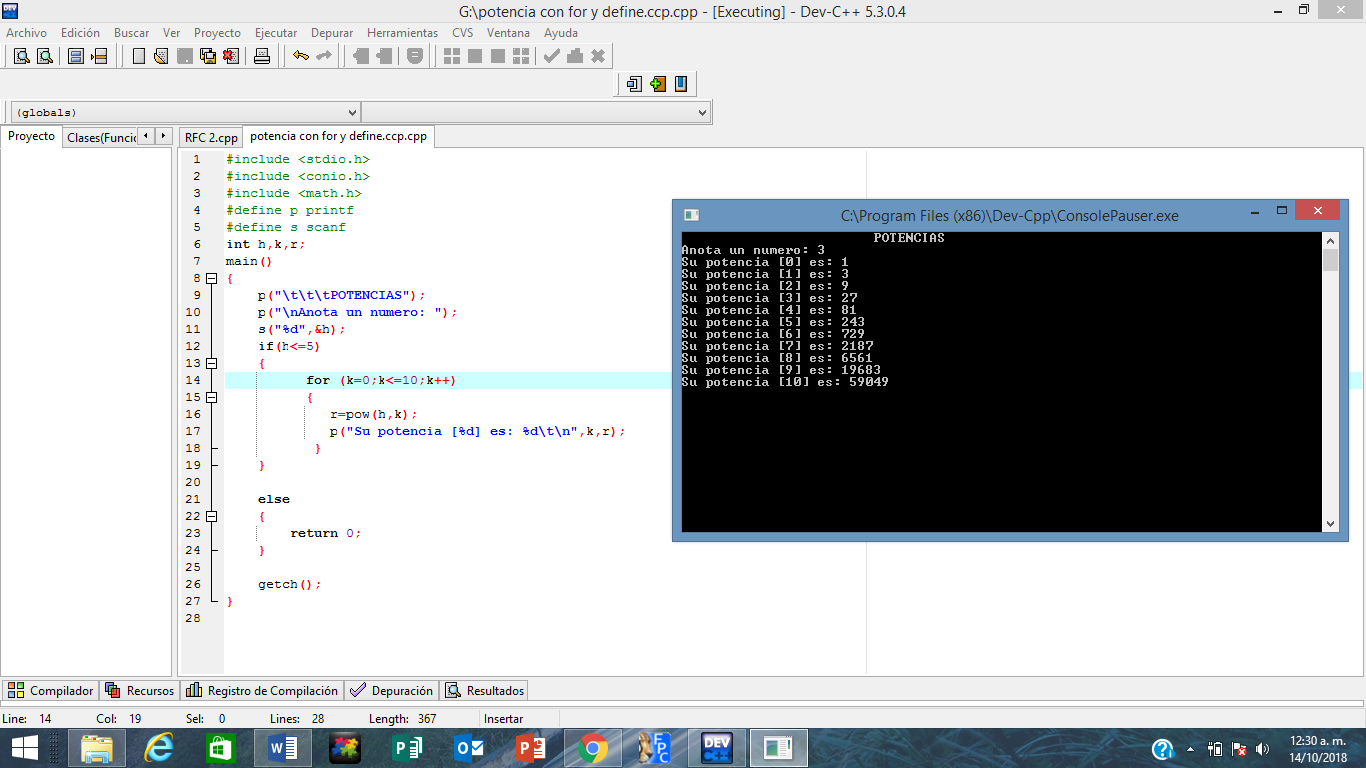
ACTIVIDAD 1: Imprimir los numeros del uno al 100 utilizando las estructuras de control while, do while y for.

En los dos primeros casos la sintaxis usada por while y do while es casi la misma,sin embargo la posicion de while cambia y se observa que para ambos casos primero se imprime el valor tomado por la variable y posteriomente se efectua el incremento para regresar a la expresion declarada en while y verificar si la condicion se sigue cumpliendo. En cambio en la estructura for la posicion inicial y la final ademas del incremento/decremento son escritos en una misma linea y la funcion printf se escribe entre los corchetes para permitir que todos los valores se impriman



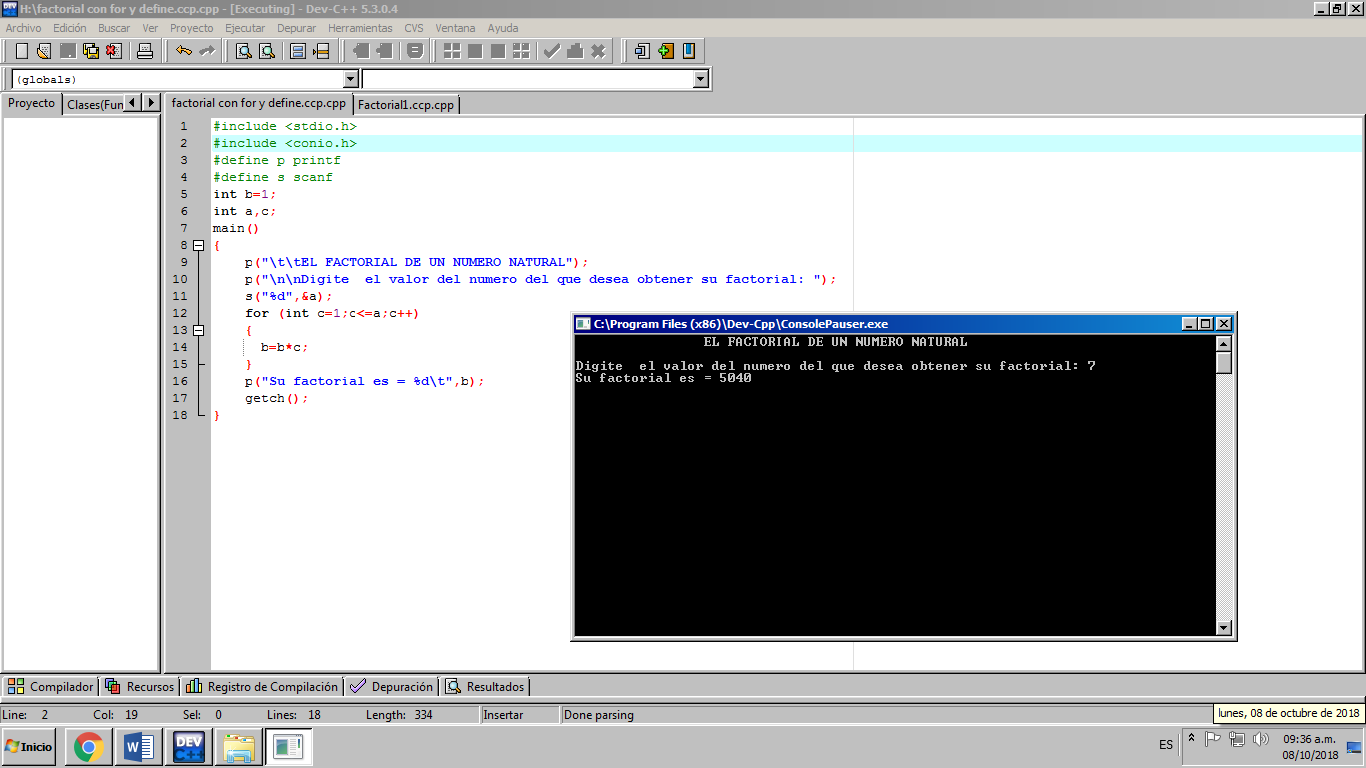
ACTIVIDAD 2: “Las primeras 10 potencias de cualquier numero en el intervalor del 1 al 5”

Para este ejercicio no fue necesario utilizar aun un for anidado, en su lugar utilizamos un if para definir que solamente se aceptaban valores entre las cifras numericas del 1 al 5 y un for para multiplicar de una 10 veces el mismo numero.



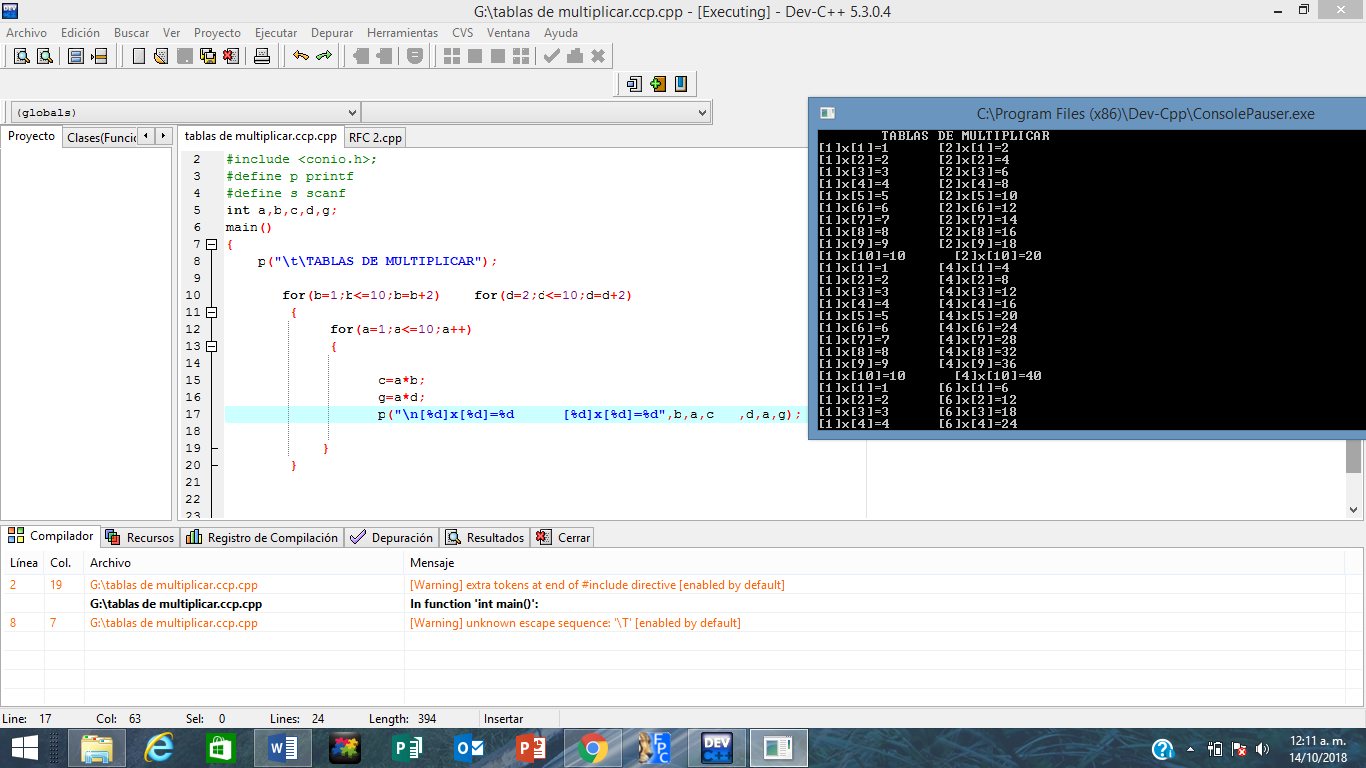
ACTIVIDAD 3: “El factorial de un número”

Este es un programa que ya se había realizado anteriormente, usa la estructura for y se detiene al momento en el que el la variable=1 que controla el número de veces que esta debe ser multiplicada es igual a la variable que representa al número del cual se desea obtener su factorial; además de esto, lo único que cambia en este programa fue la declarativa define que también aparece en los programas anteriores, la cual tuvo una función muy práctica para abreviar las funciones printf y scanf.



ACTIVIDAD 4: “TABLAS DE MULTIPLICAR”

Este último es un programa sencillo que usa un par de for anidados, uno para describir el número de tablas de multiplicar que se van a obtener, y el segundo para controlar la cantidad de cifras numéricas por las que se va a multiplicar cada tabla. El procedimiento es bastante sencillo mas no es así el caso del acomodo de posiciones que se me complico al momento de querer separar las tablas en dos columnas distintas.



CONCLUSION: Hay que recordar perfectamente el funcionamiento de cada estructura para manejarlas correctamente y así evitar los ciclos infinitos de los procesos a realizar. Y otro aspecto no menos importante es el uso de los for anidados, que además de ser utilizados como lectores de variables en los arreglos pueden realizar n veces la misma operación del otro for que los contiene; por ello se debe tener cuidado con ellos, pues, se podría tener un conjunto de repeticiones mayores a las esperadas del programa desarrollado.