**ANALISIS CICLO FOR Y CICLO FOR ANIDADO.**

**CICLO FOR.**

Como se sabe, el ciclo **for** permite ahorrar tiempo de codificación pues a diferencia del ciclo **do** – **while** y el ciclo **while** en los que deben tener cuidado al inicializar la variable que incrementará, así como no olvidar realizar el incremento de la misma, con el **for** este proceso se hace más fácil pues la estructura ya trae consigo este proceso.

El funcionamiento del **for** es diferente a los anteriores (aunque su lógica sea la misma):

1. La primera vez que se ingresa al ciclo se inicializa la variable
2. Se evalúa la condición, si esta arroja verdadero, ingresa y ejecuta los procesos, sino el ciclo se rompe y continúa con el algoritmo sin entrar en él.
3. al terminar de ejecutar los procesos se realiza el incremento de la variable.
4. Se regresa al paso 2 hasta que la condición arroje falso y se salga.

Hay que tener en cuenta que el paso 1 solo se realiza una vez, después cuando se ejecuta la primera iteración del ciclo, la variable no se vuelve a inicializar.

**Ejemplo.**

El siguiente algoritmo permite imprimir la tabla de un número dado.



Para este caso nótese que el número de la tabla se almacena en la variable num siendo num=5, por lo tanto como el ciclo se repetirá 10 veces, se utilizan los valores del contador del ciclo para que este se multiplique por el numero dado.

Aquí el for nos permite reutilizar cada vez el contenido de la variable i, pues en cada iteración este incrementa de uno en uno pudiendo ser multiplicado por el num y arrojando así la tabla deseada.

**CICLO FOR ANIDADO.**

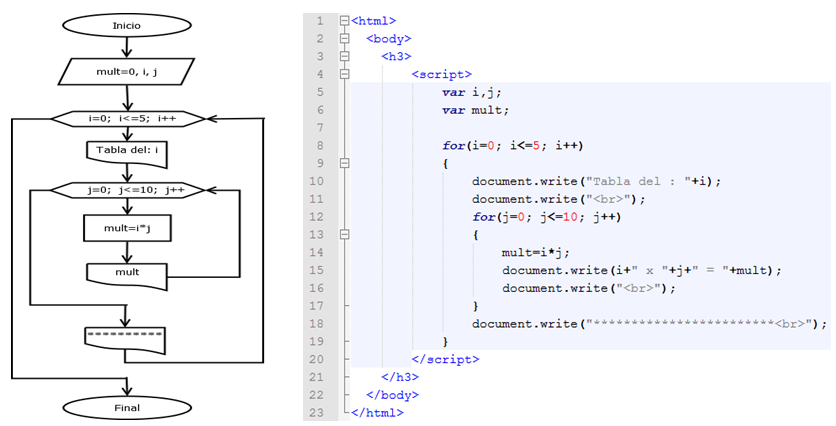
Cuando hablamos de un proceso anidado, nos referimos a la forma de unir procesos dentro de otros, si anidamos un if, quiere decir que dentro de esta estructura puede haber otro if y así sucesivamente.

Esta lógica se aplica a cualquier estructura, por esa razón los ciclos no están exentos de ella. Cuando usamos ciclos anidados es porque necesitamos repetir un proceso largo varias veces, hasta el momento tan solo usamos ciclos para casos puntuales donde hay ciertas iteraciones controladas fácilmente por una condición de parada, pero y si queremos que un proceso que se repite muchas veces, sea repetido también muchas veces? En este caso usamos ciclos anidados para que el ciclo más interno se encargue de repetir los procesos puntuales, y el ciclo externo se encargue de repetir nuevamente el ciclo interno hasta que el externo termine.

Veamos el mismo ejemplo anterior donde deseamos encontrar la tabla de un número, pero en este caso queremos encontrar las tablas desde el 0 hasta el 5.

Implícitamente cuando hablamos de obtener la tabla de un número, por lo regular pensamos en la multiplicación de ese número por todos los números del 0 hasta el 10, en ese caso tan solo pensamos en un solo ciclo, pero cuando ya nos hablan de varias tablas, veremos que ese proceso que hicimos para uno, tiene que repetirse para todos los números que deseen.

El ejercicio habla de los números del 0 al 5, como alternativa podríamos hacer varios For con la misma lógica, uno por cada número, pero ese proceso es poco optimo al tener que duplicar el mismo código cada vez, por esa razón mejor pensamos en hacer un for anidado, donde el ciclo externo se encargue de controlar para que número calcularemos la tabla y el interno se encargue de calcular dicha tabla… veamos.



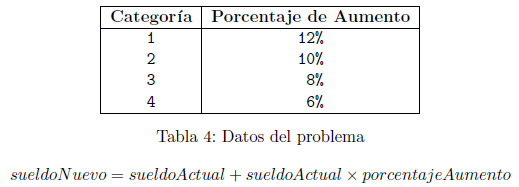
Como vemos podemos darnos cuenta que tenemos el ciclo externo controlado por la variable i que representará el número de la tabla, y tenemos el ciclo interno con la variable j que representa los números del 0 al 10 por los cuales i será multiplicado. Cuando el ciclo interno termina, le da paso al ciclo externo para que se repita y aumente al siguiente número, al hacerlo el ciclo interno volverá a empezar cada vez.

Para entender mejor este proceso, estudie y complete la siguiente prueba de escritorio.

|  |  |
| --- | --- |
| Iteración ciclo externo cuando **i vale 0** el ciclo interno se repite hasta que se cumple la condición haciendo que **j** incremente cada vez de uno en uno hasta 10 |  |
| Iteración ciclo externo cuando **i vale 1** el ciclo interno se repite hasta que se cumple la condición haciendo que **j** incremente cada vez de uno en uno hasta 10 |  |
| Iteración ciclo externo cuando **i vale 2** el ciclo interno se repite hasta que se cumple la condición haciendo que **j** incremente cada vez de uno en uno hasta 10 |  |

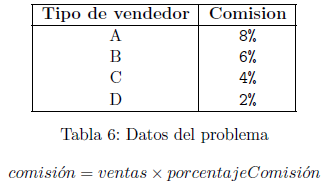
**EJERCICIOS.**

**1.** Dados el sueldo actual y la categoría de un empleado, calcular el nuevo sueldo si se efectuara un aumento de acuerdo con la tabla 4.



Para resolver el ejercicio, deberá crear un menú de opciones con las diferentes categorías, adicional una de las opciones será la opción salir, si el usuario selecciona está el sistema finalizará, de lo contrario el sistema seguirá presentando el menú y disponible para realizar el cálculo.

**2.** Calcular el pago de la comisión sobre las ventas a un vendedor de acuerdo al tipo de vendedor como está en la tabla 6



Igual que el ejercicio anterior, el sistema deberá repetir el proceso hasta que el usuario indique que quiere salir.

**3.** Un agricultor desea saber el costo de producción de un cultivo de acuerdo a la semilla seleccionada, conociendo el número de semillas que utilizará y el costo de cada semilla. El valor de cada semilla se encuentra en la tabla 11.

|  |  |
| --- | --- |
| Para resolver este ejercicio analice el enunciado, note que el usuario deberá seleccionar un cultivo y dependiendo del cultivo deberá seleccionar el tipo de semilla que quiere sembrar, por lo tanto tiene que tener en cuenta que deberá presentar un nuevo menú dependiendo de la opción elegida y con base a esto realizará los cálculos correspondientes.  Adicionalmente este proceso se realizara hasta que el usuario desee finalizarlo. |  |

**4.** Programa que genere los números primos menores que 100. Un número n es primo sı y solo si sus dos únicos divisores son 1 y n. (Ej: 1,2, 3, 5, 7, 11, 13, 17… 97)

**5.** Hallar los divisores de un entero positivo dado.

**6.** Dados a y b, enteros positivos, calcular a x b usando únicamente sumas sucesivas.

**7.** Hallar los divisores de un entero positivo dado por el usuario, este proceso lo deberá realizar las veces que el usuario decida, para esto debe usar ciclos anidados.

**8.** Dado un número ingresado por un usuario, deberá imprimir su tabla, este proceso lo deberá realizar las veces que el usuario decida, para esto debe usar ciclos anidados.

**9**. Haga un algoritmo que solicite el nombre y la edad de una persona y luego imprima un saludo como por ejemplo: “Hola pepe, su edad es 18”. Valide que las edades no sean negativas, en caso de serlo, el sistema deberá pedir nuevamente la edad hasta que ingrese un valor positivo y permita imprimir el saludo, adicionalmente este proceso debe realizarlo para 5 personas.

**7.** Crear un rectángulo en la pantalla con el carácter \*. El usuario ingresará el ancho y el alto, para esto deberá usar 2 ciclos anidados.

