

TEMA: ESTRUCTURA DE CONTROL REPETITIVA (SESION 06)

SEMESTRE 2024-1

LOGRO DEL APRENDIZAJE:

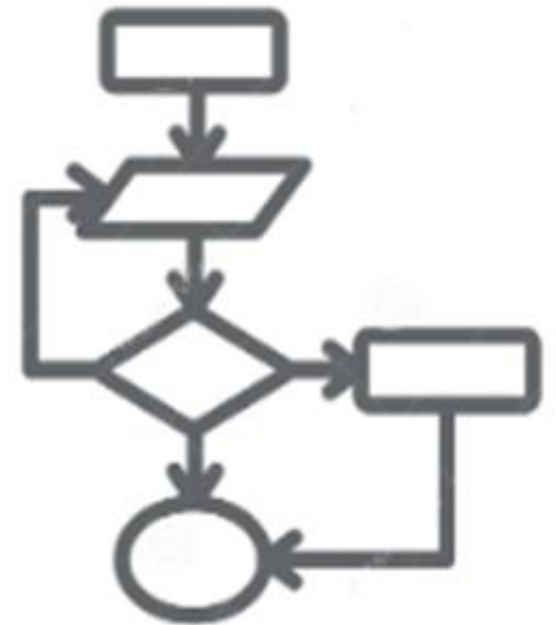
El estudiante aplica las estructuras de control selectivas y repetitivas en la resolución de problemas.

KEYWORDS:

Estructura repetitiva, programa.

INDICE

- Estructura de Control Repetitiva
 - ✓ Sintaxis de la instrucción Do while
 - ✓ Sintaxis de la instrucción For
 - ✓ Ejemplos de do while
 - ✓ Ejemplos de for
 - ✓ Ejemplos de do while y for



2.2 ESTRUCTURA DE CONTROL REPETITIVA

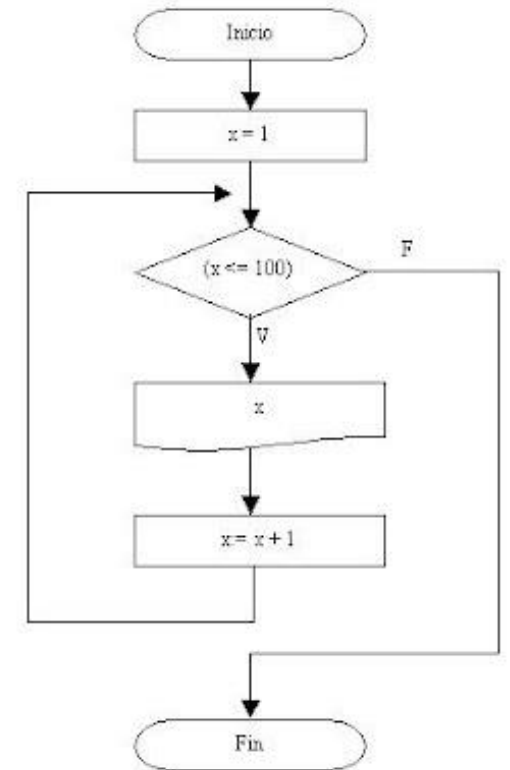
SENTENCIA REPETITIVA

Sentencias de programación que permite resolver un problema donde debemos repetir un conjunto de pasos hasta que una condición se cumpla.

Son aquellas que permiten repetir la ejecución de un conjunto de instrucciones siempre y cuando una condición se cumpla, a la entrada o a la salida del LOOP o LAZO.

Tipos:

- ✓ do while
- ✓ for
- ✓ while



2.2 ESTRUCTURA DE CONTROL REPETITIVA

SENTENCIA REPETITIVA DO WHILE (hacer ... mientras)

Se hace o ejecuta el bloque de instrucciones mientras la condición sea VERDADERA. Si la condición es FALSA se da por terminado el ciclo repetitivo



Siempre se ejecuta el bloque de instrucciones por lo menos una vez.

Si el bloque de instrucciones sólo contiene una instrucción, no es necesario colocar {}

```
do {  
    bloque de instrucciones  
} while (condición);
```

2.2 ESTRUCTURA DE CONTROL REPETITIVA

SENTENCIA REPETITIVA FOR (para ...)

Se ejecuta el bloque de instrucciones mientras la condición sea VERDADERA. Si la condición es FALSA se da por terminado el ciclo repetitivo

- a) Se ejecuta la inicialización.(POR UNICA VEZ)
- b) Se analiza la condición.
- c) Se ejecuta el bloque de instrucciones. (SOLO si la condición es TRUE)
- d) Se incrementa o decrementa y se vuelve al paso 2 .
- e) Se sale del for (SOLO si la condición es FALSE)

Si el bloque de instrucciones sólo contiene una instrucción, no es necesario colocar {}

```
for (inicialización; condición; incremento)
{
    bloque de instrucciones
}
```



2.2 ESTRUCTURA DE CONTROL REPETITIVA

EJEMPLO DE SENTENCIA REPETITIVA DO WHILE

El DO WHILE se usa en ocasiones para poder validar los datos de entrada y asegurarnos que no ingresen datos incorrectos.



VALIDAR UN NUMERO	VALIDAR UN CARÁCTER
<pre>float nota; do { cout<<"Ingrese nota: "; cin>> nota; }while(nota<0 nota>20);</pre>	<pre>char talla; // S M L do{ cout<<"Ingrese talla (S M L): "; cin>> talla; }while(talla!='S' && talla!='M' && talla!='L');</pre>

Si Ud. ingresa una nota negativa o una mayor a 20, el DO WHILE va obligar a pedir de nuevo la nota.

Si Ud. ingresa una talla diferente a S y M y L, el DO WHILE va obligar a pedir de nuevo la talla

2.2 ESTRUCTURA DE CONTROL REPETITIVA

EJEMPLO DE SENTENCIA REPETITIVA DO WHILE



Desarrollar un programa en C++ usando estructura de control repetitiva, que promedie los pesos de los pacientes hasta que se ingresen un peso -1 como condición de fin de datos.

CONDICION DE FIN DE DATOS

```
void main()
{ float peso,suma=0,cont=0;
  do{
    cout<<"Ingrese peso: ";
    cin>> peso;
    if(peso != -1) {
      suma=suma+peso;
      cont++;
    }
  }while (peso!=-1);
  cout<<"El promedio es : "<<suma/cont;
  _getch();
}
```


2.2 ESTRUCTURA DE CONTROL REPETITIVA

EJEMPLO DE SENTENCIA REPETITIVA DO WHILE

Desarrollar un programa en C++ usando estructura de control repetitiva, que realice las 4 operaciones aritméticas básicas con 2 números enteros usando un menú, hasta que se ingresen un valor 5 como condición de fin de datos



CONDICION DE FIN DE DATOS

```
void menu()
{ cout << endl << "-----" << endl;
  cout << "Menu " << endl;
  cout<<"1. Suma "<<endl;
  cout<<"2. Resta "<<endl;
  cout<<"3. Multiplica "<<endl;
  cout<<"4. Divide "<<endl;
  cout<<"5. Salir "<<endl;
}
```

```
void main()
{ int opc,num1,num2;
  do{menu();
    cout<<"Ingrese opcion : "; cin>> opc;
    if (opc!=5)
    { cout<<"Ingrese 2 enteros : "; cin>> num1>>num2;
      switch(opc)
      { case 1: cout<<"La suma es : "<<num1+num2;break;
        case 2: cout<<"La resta es : "<<num1-num2;break;
        case 3: cout<<"La multiplicación es : "<<num1*num2;break;
        case 4: cout<<"La division es : "<<num1/num2;
      }
    }
  }while (opc!=5);
}
```

2.2 ESTRUCTURA DE CONTROL REPETITIVA

EJEMPLO DE SENTENCIA REPETITIVA FOR

Desarrollar un programa en C++ usando estructura de control repetitiva



Que imprima los 10 primeros números naturales	Que sume los 10 primeros números naturales
<pre>void main() { for (int i=1;i<=10;i++) cout<<i<<' '; _getch(); }</pre>	<pre>void main() { int suma=0; for (int i=1;i<=10;i++) suma=suma+i; cout<<"suma es :"<<suma; _getch(); }</pre>

2.2 ESTRUCTURA DE CONTROL REPETITIVA

EJEMPLO DE SENTENCIA REPETITIVA FOR

Desarrollar un programa en C++ usando estructura de control repetitiva



Que cuente los impares entre los 20 primeros números naturales (Forma 1)	Que cuente los impares entre los 20 primeros números naturales (Forma 2)
<pre>void main() { int cont=0; for (int i=1;i<=20;i++) if (i% 2 != 0) cont++; cout<<"impares hay :"<<cont; _getch(); }</pre>	<pre>void main() { int cont=0; for (int i=1;i<=20;i+=2) cont++; cout<<"impares hay :"<<cont; _getch(); }</pre>

2.2 ESTRUCTURA DE CONTROL REPETITIVA

EJEMPLO DE SENTENCIA REPETITIVA DO WHILE Y FOR

Desarrollar un programa en C++ usando estructura de control repetitiva



Que imprima los N primeros números naturales	Que sume los N primeros números naturales
<pre>void main() { int N; do{ cout<<"Ingrese N: "; cin>> N; }while(N<=0); for (int i=1;i<=N;i++) cout<<i<<' '; _getch(); }</pre>	<pre>void main() { int N; do{ cout<<"Ingrese N: "; cin>> N; }while(N<=0); for (int i=1;i<=N;i++) suma=suma+i; cout<<"suma es :"<<suma; _getch(); }</pre>

CONCLUSIONES

La sentencia repetitiva DO WHILE es una instrucción en que el cuerpo(sentencias) del bucle se repite mientras que se cumple una determinada condición. En este tipo de sentencias repetitiva, la condición del ciclo se evalúa al final, por lo que siempre se ejecutarán las instrucciones del ciclo por lo menos una vez.

La sentencia repetitiva FOR es un número total de iteraciones que se deben ejecutar una o varias sentencias o instrucciones. Esta sentencia repetitiva se caracteriza por tener un contador, una condición que involucra este contador y una sentencia de incremento o decremento dependiendo el caso.

