



Semana-03

Diseño y codificación de algoritmos repetitivos



Temario



- Uso de distintas lógicas de repetición en la construcción y codificación de algoritmos
- Contadores y acumuladores
- Generación de números aleatorios

IMPORTANCIA DE USAR ALGORITMOS REPETITIVOS



El poder real de un programa se demuestra cuando se desea repetir la ejecución de un bloque de instrucciones mas de una vez.

Casos:

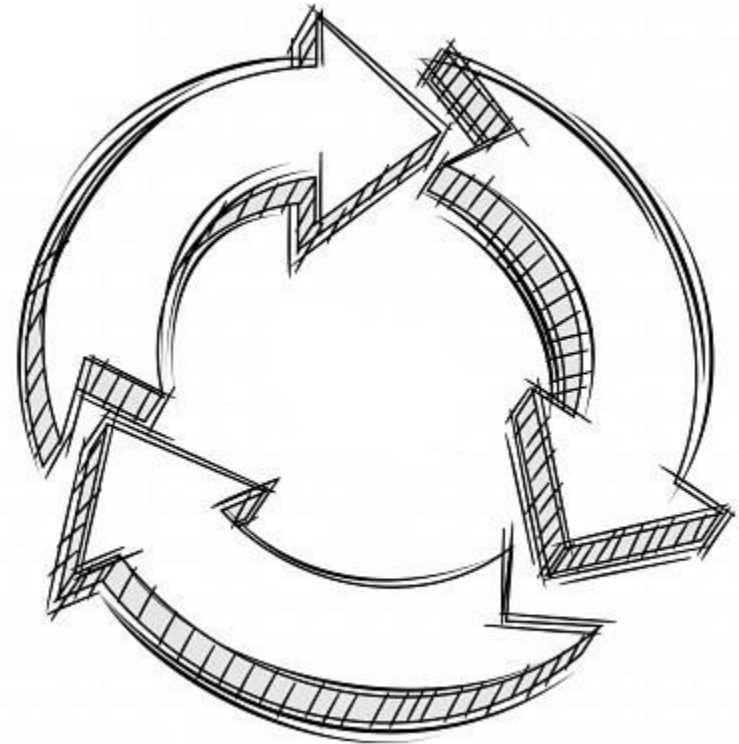
Calcular la sumatoria de un conjunto de datos.

Imprimir una serie de datos

Simplificar una fracción

.
. .
. .

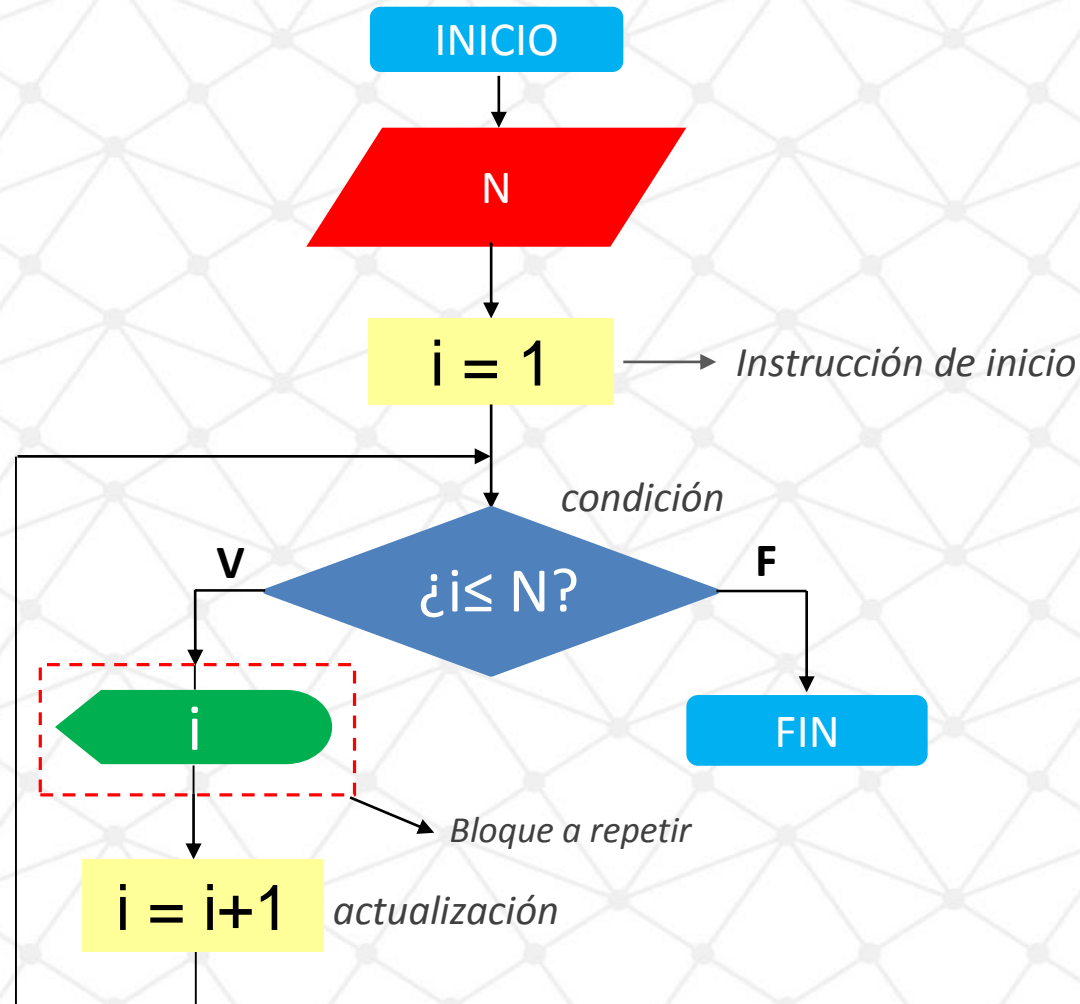
Mover un objeto en la pantalla





Ejemplo Imprimir la serie: 1 2 3 4 5 6N

Algoritmo repetitivo





EN NARRATIVO (Forma 1)

INICIO

1. Leer el valor de N
2. Asignar 1 a la variable i
3. Mientras el valor de la variable i sea \leq a N, hacer:
 4. Imprimir el valor de i en la pantalla
 5. Aumentar a la variable i el valor 1

FIN

Algoritmo listo para su codificación



EN NARRATIVO **(Forma 2)**

INICIO

1. Leer el valor de N

2. Para $i \rightarrow 1, 2, 3, 4, \dots, N$:

3. Imprimir el valor de i en la pantalla

FIN

Algoritmo listo para su codificación



Para la codificación de algoritmos repetitivos el lenguaje C++ puede utilizar 3 estructuras:

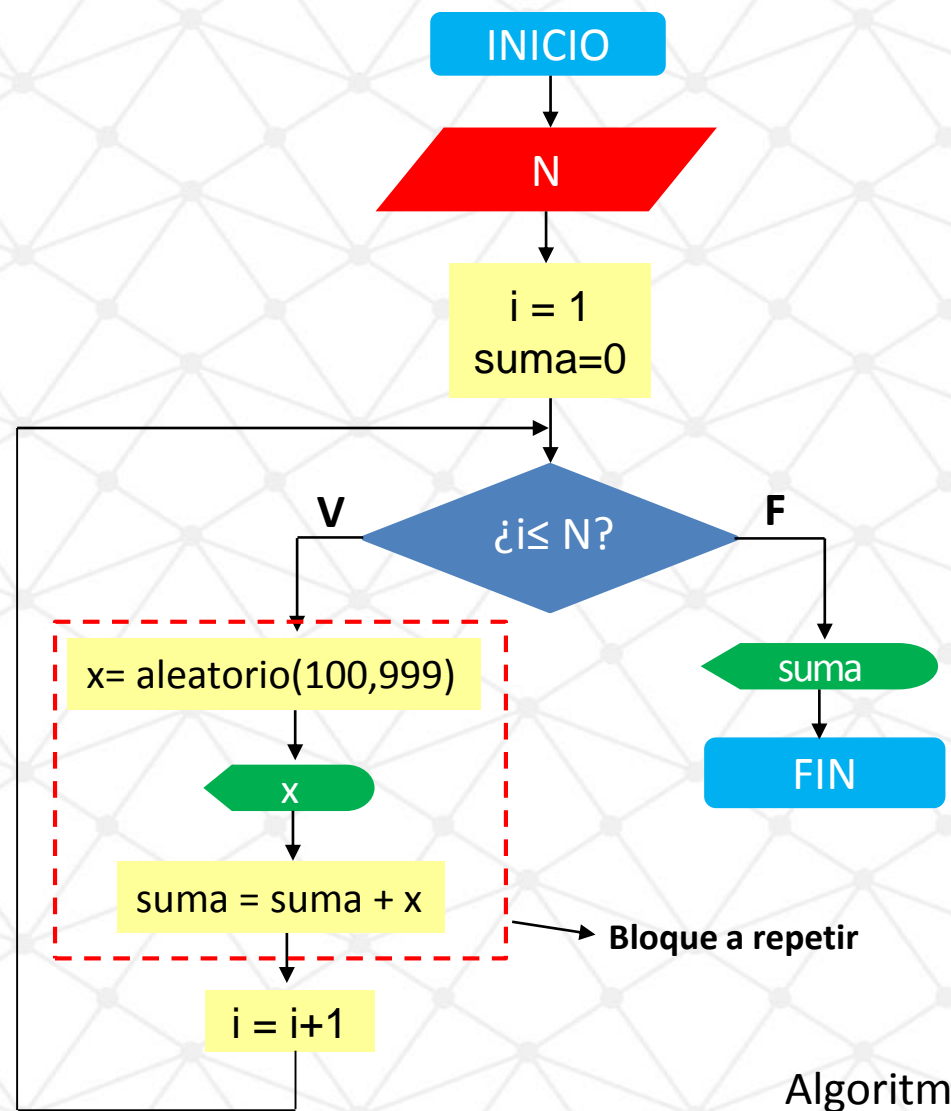
Estructura **while**

Estructura **do while**

Estructura **for**

PROBLEMA

Desarrolle e implemente un programa que permita generar y mostrar N números aleatorios de 3 cifras y al final mostrar la suma de todos ellos



Algoritmo listo para su codificación

EN NARRATIVO – (while)



INICIO

1. Leer el valor de N
2. Asignar 1 a la variable i
3. Asignar 0 a la variable suma
4. Mientras el valor de la variable i sea \leq a N, hacer:
 5. Asignar a la vble. x un aleatorio en el rango [100, 999]
 6. Imprimir el valor de la vble. x
 7. Aumentar a la variable suma el valor de x
 8. Aumentar a la variable i el valor 1

FIN

Algoritmo listo para su codificación



INICIO

1. Leer el valor de N
2. Asignar 0 a la variable suma
3. Para $i \rightarrow 1, 2, 3, 4, 5, \dots, N$:
 4. Asignar a la vble. x un aleatorio en el rango [100, 999]
 5. Imprimir el valor de la vble. x
 6. Aumentar a la variable suma el valor de x

FIN

Algoritmo listo para su codificación



```
float a,b,c,disc,x1,x2;  
char x='S';
```

```
while(x == 'S')  
{
```

BLOQUE

Algoritmo codificado en C++ que
resuelve una ecuación cuadrática.
(desarrollado en la semana-02)

```
cout<<endl<<"Deseas continuar S/N: ";  
cin>>x;  
}
```

Después que el algoritmo resuelve la ecuación cuadrática, se le preguntará al usuario si desea continuar o no. En caso ingrese el carácter 'S' (SI) se repetirá nuevamente el algoritmo, en caso ingrese el carácter 'N' (NO) el programa termina