

---

# ESTRUCTURAS REPETITIVAS

---

**PROGRAMACION II**

- Necesarias en la mayoría de los algoritmos.
- Estructura repetitiva = bucle = lazo.
- Iteración
- Ejecutan un conjunto de acciones(cuerpo del bucle) un número *determinado* o *indeterminado* a priori de veces.
- ¿Cuántas veces se ejecutan las acciones?

Condición de control

## *A) Número de veces indeterminado a priori.*

### Estructura **MIENTRAS**

- *Sintaxis:*

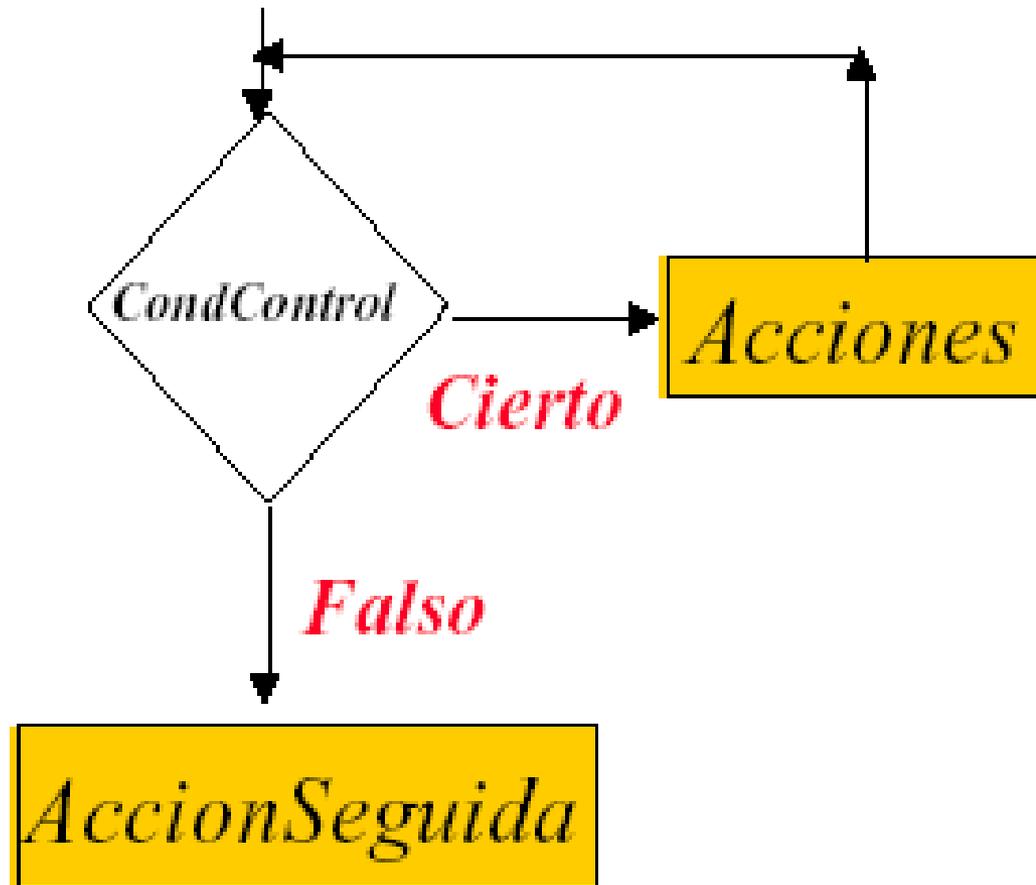
**MIENTRAS** CondControl **HACER**

Acciones

**FINMIENTRAS**

AcciónSeguida

# Diagrama de flujo



- Condición de control del bucle = expresión lógica
- Las sentencias que forman el cuerpo del bucle se ejecutan **0 o más veces**.
- BUCLE **PREPRUEBA**.
- Condición *CIERTO*  seguirá ejecutando el cuerpo
- Algunos ejemplos de bucles :

*Bucle controlado por contador*

*Bucle controlado por centinela*

*Bucle contador*

## *Bucle controlado por contador*

- Se ejecuta un número determinado de veces.
- Variable de control del bucle

inicialización, evaluación, incremento

```
i ← 1          /*Inicialización*/
```

```
MIENTRAS i < 10 HACER /*Evaluación*/
```

```
    acciones
```

```
    i ← i + 1      /*Incremento*/
```

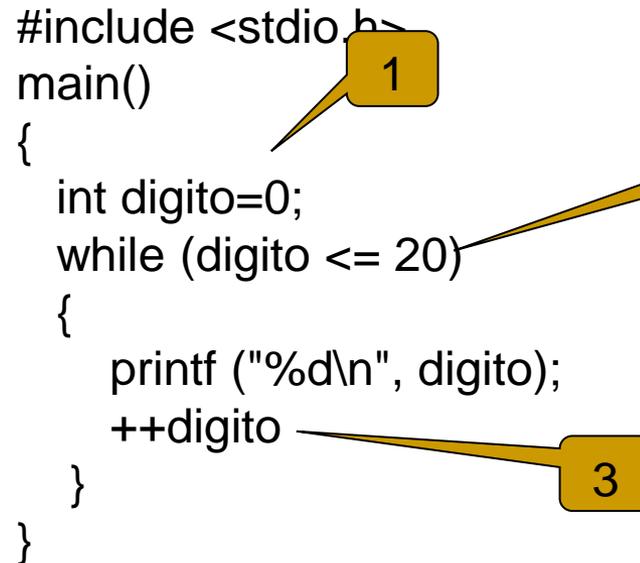
```
FINMIENTRAS
```

- ¿Cuántas veces se ejecuta el anterior bucle?

# while (expresión) sentencia

## Ejemplo 1

```
#include <stdio.h>
main()
{
    int digito=0;
    while (digito <= 20)
    {
        printf ("%d\n", digito);
        ++digito
    }
}
```



## Ejemplo 2.

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

```
/*Calcular la media de n números*/
```

```
main()
```

```
{
```

```
int n, cont=1;
```

```
float x, media, suma=0;
```

1

```
clrscr();          /* inicializa y leer el valor de n */
```

```
printf ("Cuantos números va a promediar? \n");
```

```
scanf ("%d",&n);
```

```
while (cont <= n)
```

```
{          /* Leer los números */
```

```
printf ("X = ");
```

```
scanf ("%f",&x);
```

```
suma+= x;
```

```
++cont;
```

```
}
```

2

3

```
/* Calcular la media y escribir la respuesta*/
```

```
media = suma/n;
```

```
printf ("\n La media es %4.3f\n",media);
```

```
delay (5000); }
```

Un generador de secuencias es capaz de generar una lista de números hasta un límite configurable, partiendo de un valor de inicio y siguiendo un incremento también configurable. Así, por ejemplo, especificando como valor de inicio el número 1, como incremento 2, y como valor final 100, el dispositivo genera los números impares menores que 100. De igual forma se podrían generar secuencias de números pares, u otras secuencias con incrementos mayores. Incluso, es posible generar secuencias decrecientes, especificando un valor de incremento que sea negativo. Diseñe un programa que implemente un generador de secuencias.

## ANALISIS DEL PROBLEMA

Inicio = 1

Incremento = 2

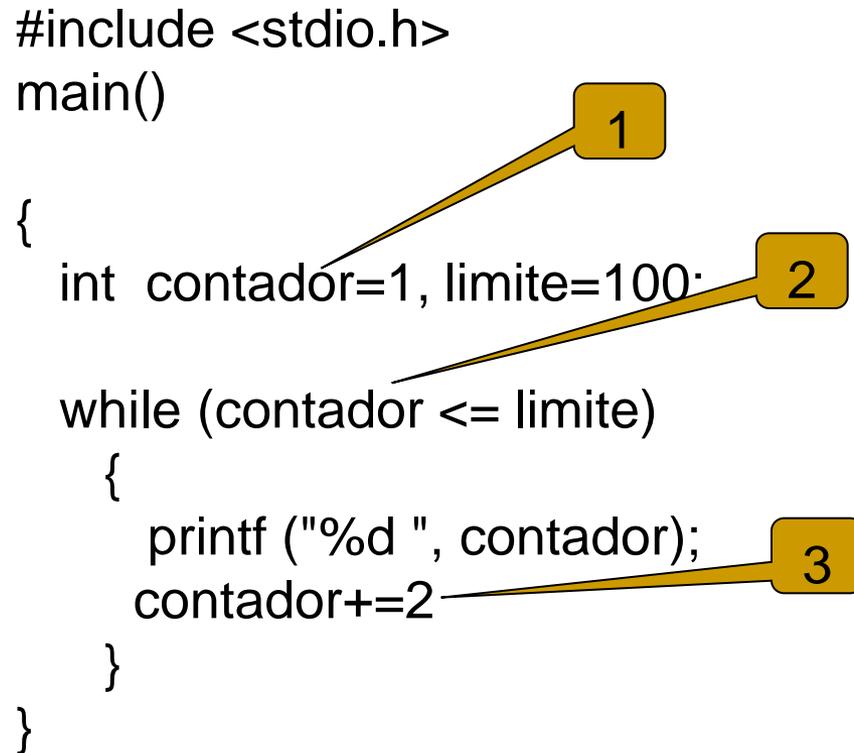
Valor final = 100

SECUENCIA:

1, 3, 5, 7, 9, 11, 13, 15, 17, ... 99

### EJEMPLO # 3

```
#include <stdio.h>
main()
{
  int contador=1, limite=100;
  while (contador <= limite)
  {
    printf ("%d ", contador);
    contador+=2;
  }
}
```



1

2

3

Una fábrica de cuadernos requiere un programa para generar tablas de multiplicar que puedan ser impresas en las tapas de los cuadernos que fabrica. Dado que los cuadernos tienen distintas dimensiones, se requiere que el programa sea configurable, para poder generar distintos tipos de tablas de multiplicar, para distintos números y llegando hasta límites variables. Por ejemplo, la tabla del 5, llegando hasta 12, es decir, que incluya todos los múltiplos de 5 desde  $5 \times 1$  hasta  $5 \times 12$ . Para que sea útil, la tabla no solo debe mostrar el resultado, sino también cómo se llegó a él, por ejemplo,  $5 \times 1 = 5$ ,  $5 \times 2 = 10$ , . . . , y así sucesivamente.

## ANALISIS DEL PROBLEMA

Leer el Numero

Calcular Numero x 1

Que se incrementa?

Numero x 2

Cuanto se incrementa?

Numero x 3

Límite de la tabla = numero x 12

...

Numero x 12

## EJEMPLO # 4

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

```
main()
```

```
{
```

```
int contador=1, numero, producto, limite=12;
```

```
clrscr();
```

```
printf("Escriba el número, por favor ");
```

```
scanf ("%d",&numero);
```

```
printf(" Tabla de multiplicar del %d \n",numero);
```

```
while (contador <= limite)
```

```
{
```

```
    producto=contador*numero;
```

```
printf ("%d x %d = %d\n ", numero,contador,producto);
```

```
contador++;
```

```
}
```

```
getch();
```

```
}
```

1

2

3

# La sentencia FOR

## Formato

FOR (inicialización, condición, incremento) sentencia

Es la variable de control índice que controla la repetición del ciclo.

Representa una condición que debe ser satisfecha para que continúe la ejecución del ciclo. El ciclo se mantiene en función mientras la condición sea verdadera.

Incremento. Modifica el valor del parámetro inicial incrementando la variable de control.

Generalmente la *inicialización* es una expresión de asignación; *condición* es una expresión lógica e *incremento* es una expresión monaria.

## EJEMPLO 5

```
#include <stdio.h>      /* visualiza los números del 0 al 9 */
main()
{
    int numero;

    clrscr();
    for (numero=0; numero <= 9; ++numero)
        printf ("%d\n ", numero);
    getch();
}
```

## EJEMPLO 6

```
#include <stdio.h>
main()
{
    int numero,suma=0;
    clrscr();
    for (numero=1; numero <= 100; numero++)
    {
        suma= suma + numero;
        printf ("%d\n", suma);
    }
    getch();
}
```

## EJEMPLO 7

```
#include <stdio.h>    /*genera los números del 50 al 1 */
main( )
{
    int numero;
    clrscr();
    for (numero=50;numero>0;numero--)
        printf ("%d ", numero);
    getch();
}
```

## EJEMPLO 8

```
#include <stdio.h> /*genera los números del 0 al 100 de 5 en 5 */
main()
{
    int numero;
    clrscr();
    for (numero=0;numero<=100;numero=numero+5)
        printf ("%d ", numero);
    getch();
}
```

## EJEMPLO 9

```
#include <stdio.h>
int sum;
main( )
{
    int cant, cont;
    float num, media, sum= 0;
    clrscr( );
    printf ("Cuantos numeros?\n");
    scanf ("%d",&cant);
    for (cont=1; cont <=cant; ++cont)
    {
        printf ("num = ");
        scanf ("%f", &num);
        sum =sum+num;
    }
    media = sum/cant;
    printf ("\n la media es %f\n", media);
    getch();
}
```