



UNIDAD 2.- ESTRUCTURA ELEMENTAL DE DATOS. DISEÑO DE ALGORITMOS.

OBJETIVOS:

- 2.1.** Que el alumno comience a formularse y resolver problemas, diseñando las estrategias correspondientes de manera clara, sistémica y por sobre todo sencilla, mediante el diseño de algoritmos y diagramas de flujo.
- 2.2.** Que el alumno logre aumentar la capacidad de reflexión del alumno, reforzando las conductas logradas mediante la Unidad 1.-

TEMAS:

- 2.1.** La implementación de las estructuras repetitivas.

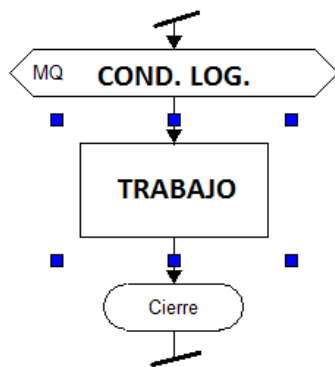
2.1. LA IMPLEMENTACIÓN DE LAS ESTRUCTURAS REPETITIVAS.

ESTRUCTURAS DE REPETICIÓN

Estos tipos pueden corresponder a varias formas de realización.

Estructura **mientras**

Diagrama de flujo

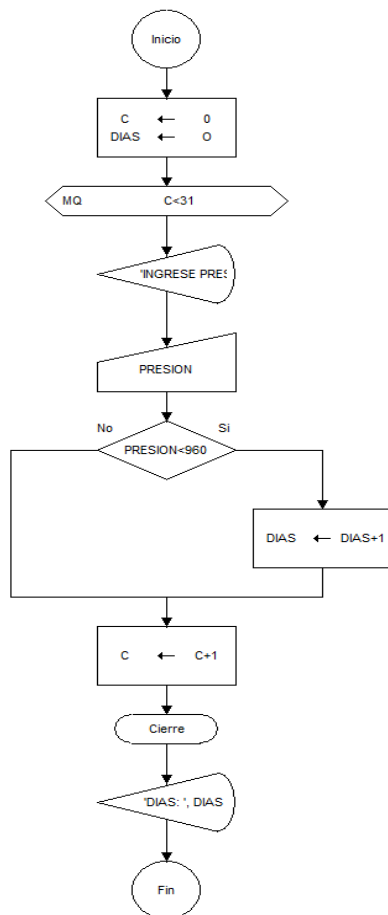


Codificación:

```
while (condicion)
{
    acción 1;
    acción 2;
    acción 3;
    .....
    acción n;
}
```

Ejemplo

Ingresar las presiones atmosféricas medidas durante 31 días de un mes del año. Contar cuantos días hubo probabilidad de lluvias o mal tiempo. (Estas condiciones se producen cuando la presión atmosférica es menor que 960 m bares).



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

```
int cont, dias;
float presion;
```

```
main()
{
    c = 0 ;
    dias = 0 ;

    while (c < 31)
    {
        printf("Ingrese presion atmosferica: ");
        scanf ("%f" , &presion ) ;

        if (presion < 960)
            dias = dias + 1 ;

        c = c + 1 ;
    }

    printf("Cantidad de días: %d", dias);
    getch();
}
```

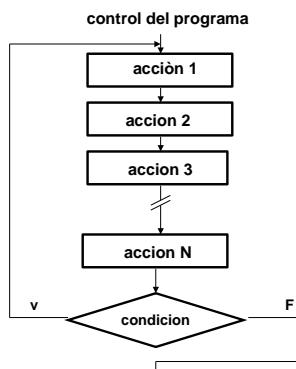
Observación:

Después de escribir la sentencia cuando la condición es verdadera, debe escribirse un punto y coma (;). También, si existe una sola instrucción cuando la condición es verdadera, se la puede escribir sin delimitadores (como en el ejemplo). En este caso, la única sentencia a ejecutarse será la siguiente a la sentencia **if**.

Estructura hacer- mientras

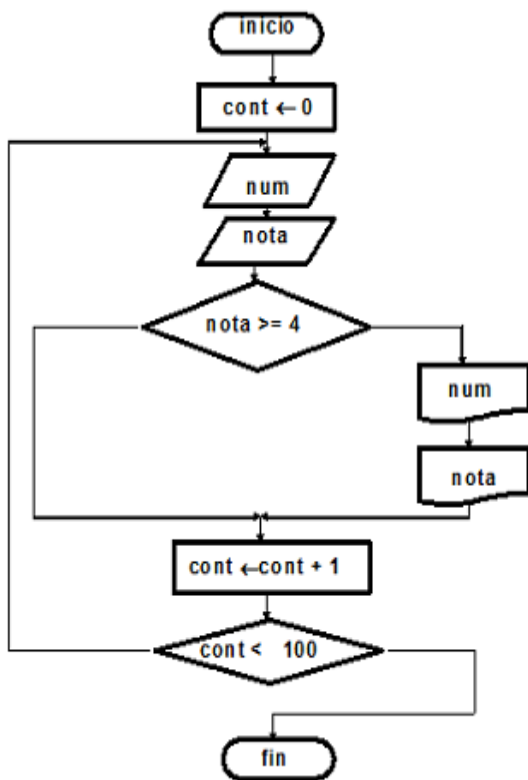
Se debe tener en cuenta que esta estructura repite por el valor falso de la condición. Para poder implementarla en Lenguaje C, se usa la estructura *Do While*, que repite siempre por la condición verdadera.

Diagrama de flujo:



Ejemplo

Ingresar el legajo (representado por un número) y la nota de 100 alumnos obtenida en un examen parcial. Mostrar el legajo y nota de aquellos alumnos que aprobaron dicho examen. El examen se aprueba con 4.



```

#include <stdio.h>
#include <conio.h>

int cont, num, nota;

main ()
{
    cont = 0;

    do
    {
        printf("\n\nIngrse el numero de legajo: ");
        scanf ("%d", &num);
        printf("Ingrese la nota: ");
        scanf ("%d", &nota);

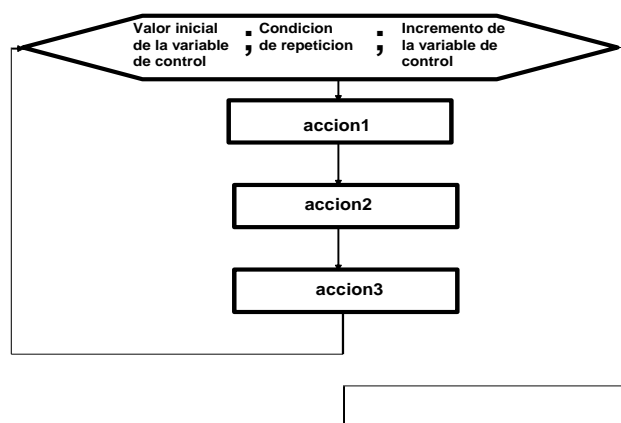
        if ( nota >= 4 )
        {
            printf ("\nLegajo de alumno aprobado: %d", num);
            printf("\nNota de aprobacion: %d", nota);
        }
        cont = cont + 1;
    }
    while (cont < 100);

    getch();
}
  
```

Estructura for

Esta estructura es equivalente a las anteriores. Su sintaxis establece que se deben escribir el valor inicial, la condición de finalización y el incremento de la variable de control en una misma directiva.

Diagrama de flujo



En Lenguaje C, en este tipo de estructura, no es necesario poner un valor inicial a una variable (de control) ya que su propia sintaxis lo especifica.

Se recomienda esta estructura cuando se conocen de antemano la cantidad total de repeticiones que realizara el lazo.

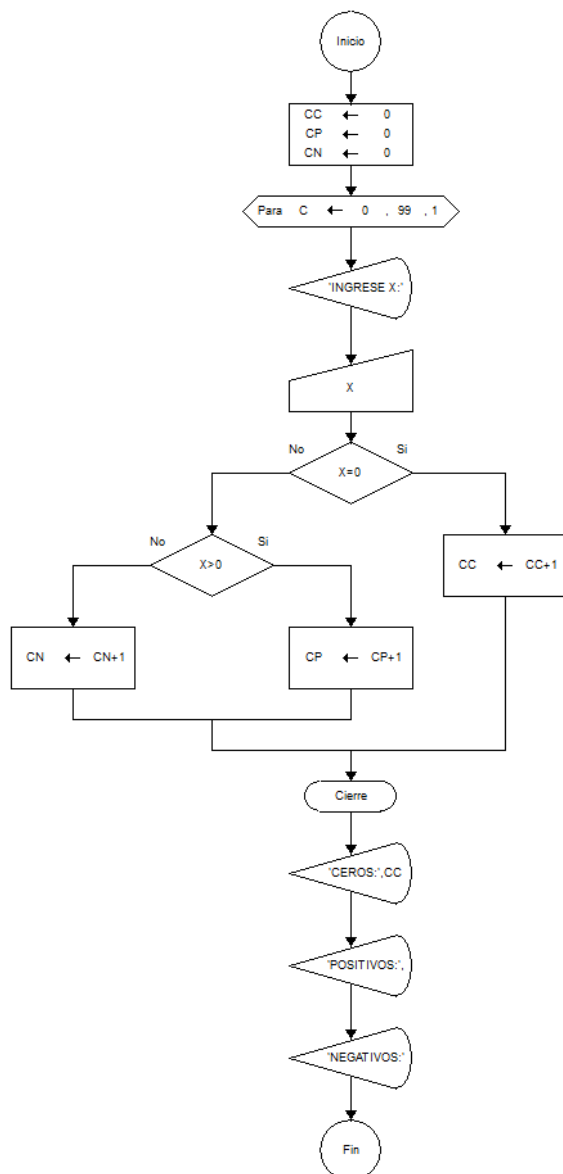
Valor inicial de la variable de control: Se indica el valor a partir del cual esta variable se incrementa con cada repetición de las acciones.

Condición de repetición del bucle: Se indica cual es la condición de terminación de las repeticiones de las acciones.

Incremento: Especifica en que valor se incrementa o decrementa la variable de control en cada iteración.

Ejemplo

Ingresar 100 números enteros y contar los positivos, los negativos y los ceros ingresados.



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

```
int x, c , cc, cp, cn;
```

```
main()
```

```
{
```

```
    cc = 0;
```

```
    cp = 0;
```

```
    cn = 0;
```

```
    for ( c = 0; c < 100; c = c + 1)
```

```
    {
```

```
        printf("Ingrese un numero: ");
        scanf ("%d" , &x);
```

```
        if (x == 0)
```

```
            cc = cc + 1;
```

```
        else
```

```
            if (x > 0)
```

```
                cp = cp + 1;
```

```
            else
```

```
                cn = cn + 1;
```

```
    }
```

```
    printf ("\n\nCantidad de ceros: %d", cc);
```

```
    printf ("\nCantidad de positivos: %d", cp);
```

```
    printf ("\nCantidad de negativos: %d", cn);
```

```
    getch();
```

```
}
```