

Laboratorio de Computación I

Sentencia DO-WHILE



Funcionamiento Sentencia Do-While – (repetir)

- En este tipo de estructura repetitiva las instrucciones interiores del bucle se repetirán **HASTA** que la condición se cumpla. Permite realizar el proceso al menos una vez, ya que la condición se evalúa al final del bucle, a diferencia del **mientras**, en el cual el proceso puede ser que nunca llegue a entrar si la condición a evaluar no se cumple desde el principio.

Sentencia Do-While – (repetir)

Do primero entra, cumple 1 vez las sentencias en el bucle y sigue cumpliendo hasta que no se cumpla la condición.

```
Do  
{  
    ....  
    ....  
    sentencias;  
    ....  
    ....  
} while (condicion);
```


Sentencia Do-While – (repetir)

Instrucción de control repetitiva : DO-WHILE

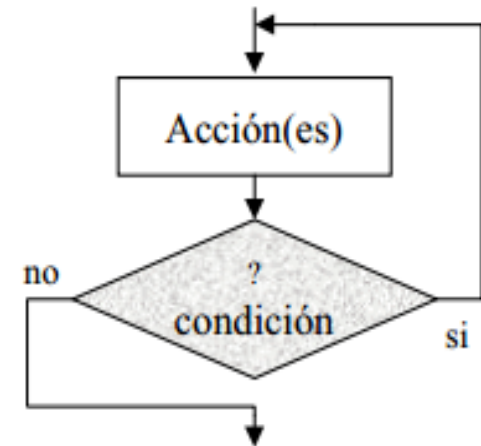
Se corresponde con la estructura algorítmica



Formato general de la
sentencia **do-while**

```
do
{
    instrucción 1;
    ...
    instrucción n;
} while ( condición );
```

```
do
    instrucción;
while ( condición );
```



Se utiliza para ejecutar un bloque de instrucciones al menos una vez.
El cuerpo del bucle **se repite mientras se verifica la condición**. Dicha condición será evaluada **después** de cada repetición.

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

```
int main()
{
    /* Escribe los numeros del 1 al 10 */
    int numero=1;

    do
    {
        printf("%d\n",numero);
        numero++;
    }
    while(numero<=10);

    printf("condicion de salida:%d\n",numero);
    getch();
    return 0;
}
```

El cuerpo de un bucle do-while se ejecuta una o más veces.

Se repite mientras la condición sea verdadera.

La condición será evaluada después de cada iteración

Ejemplo: Ingresar tres números y mostrarlos por pantalla usando DO-WHILE

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

```
int main()
{
    int cont,a;
    cont=0;

    do
    {
        printf ("Ingrese una variable");
        scanf ("%d",&a);
        printf (" Numero ingresado: %d\n ",a);
        cont=cont+1;
    }
    while (cont<3);

    getch();
    return 0;
}
```

En este problema se tiene perfectamente establecido el número de veces que un grupo de acciones se van a ejecutar. En este caso tres veces.

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

int main()
{
    int cont,a;
    cont=0;

    do
    {
        printf ("Ingrese una
variable");
        scanf ("%d",&a);
        printf (" Numero
ingresado: %d\n ",a);
        cont=cont+1;
    }
    while (cont<3);

    getch();
    return 0;
}
```

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

int main()
{
    int a,cont;

    for (cont = 0; cont<=2;cont++)
    {
        printf ("Ingrese un
valor\n");
        scanf ("%d",&a);
        printf (" Numero
ingresado: %d\n",a);
    }

    getch();
    return 0;
}
```

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

int main()
{
    int a,cont;
    Cont=0;

    While cont<2
    {
        printf ("Ingrese un
valor\n");
        scanf ("%d",&a);
        printf (" Nro.
ingresado:%d\n",a);
        cont++:
    }
    getch();
    return 0;
}
```


Ejemplo:Calcular el perímetro de un cuadrado.

Verificar que el lado ingresado sea correcto

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

```
int main()
{
    int p,a;
    p=0;

    do
    {
        printf ("Ingrese una variable positiva");
        scanf ("%d",&a);
    }
    while (a <=0);

    p= a*4;
    printf ("Perimetro=%d\n ",p);
    getch();
    return 0;
}
```

En este problema el número de repeticiones es desconocido y se hará hasta que se cumpla o no cierta condición. En este caso se repite el bucle hasta que se ingrese un valor de lado positivo.



Comparación de Bucles

WHILE	<ul style="list-style-type: none">- Adecuada para búsquedas- El cuerpo puede no ser ejecutado- La verificación de la condición precede a la ejecución del cuerpo
FOR	<ul style="list-style-type: none">- Adecuado para recorridos- La verificación de la condición precede a la ejecución del cuerpo
REPETIR	<ul style="list-style-type: none">- Adecuado en el caso de que debamos garantizar que el cuerpo del bucle se ejecuta por lo menos en una ocasión.