

# Clase V: Estructuras repetitivas while y do-while

Informática I  
Centro Regional Universitario Córdoba  
UNDEF

4 de mayo de 2021

# Estructura repetitiva while

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

# Estructura while: pseudocódigo

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

Inicio del algoritmo

# Estructura while: pseudocódigo

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

Inicio del algoritmo

Mientras (condición de ciclo==Verdadera)

# Estructura while: pseudocódigo

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

Inicio del algoritmo

```
Mientras (condición de ciclo==Verdadera)  
    acción 1;
```

# Estructura while: pseudocódigo

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

Inicio del algoritmo

Mientras (condición de ciclo==Verdadera)

    acción 1;

    acción 2;

# Estructura while: pseudocódigo

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

Inicio del algoritmo

Mientras (condición de ciclo==Verdadera)

    acción 1;

    acción 2;

    acción 3;

# Estructura while: pseudocódigo

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

Inicio del algoritmo

Mientras (condición de ciclo==Verdadera)

    acción 1;

    acción 2;

    acción 3;

...



# Estructura while: pseudocódigo

Es una estructura repetitiva controlada por centinela. Es decir, que las acciones dentro del bloque de repetición serán ejecutadas **si y sólo si** la expresión lógica evaluada en la condición de ciclo es verdadera.

Inicio del algoritmo

Mientras (condición de ciclo==Verdadera)

    acción 1;

    acción 2;

    acción 3;

    ...

    acción n;

Fin del algoritmo



```
1 #include <stdio.h>
2 int main()
3 {
4     int numero=0;
5     printf("Para salir ingrese -1\n");
6     scanf("%d",&numero);
7     while(numero!=-1)
8     {
9         printf("Ingrese un numero entero\n");
10        printf("Para salir ingrese -1\n");
11        scanf("%d",&numero);
12    }
13    printf("El ciclo repetitivo ha terminado");
14    return (0);
15 }
```

# Estructura repetitiva do-while

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

# Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

# Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

# Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

acción 1;

# Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

acción 1;

acción 2;



# Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

acción 1;  
acción 2;  
acción 3;

# Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

acción 1;

acción 2;

acción 3;

...

# Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

acción 1;

acción 2;

acción 3;

...

acción n;

# Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

acción 1;

acción 2;

acción 3;

...

acción n;

Mientras (condición de ciclo==Verdadera)

# Estructura repetitiva do-while: pseudocódigo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

Inicio del algoritmo

Hacer

acción 1;

acción 2;

acción 3;

...

acción n;

Mientras (condición de ciclo==Verdadera)

Fin del algoritmo

# Estructura repetitiva do-while: diagrama de flujo

La instrucción do...while es controlado por centinela pero evalúa la condición de continuación de ciclo **después de ejecutar el cuerpo del ciclo**; por lo tanto, el cuerpo del ciclo siempre se ejecutará al menos una vez.

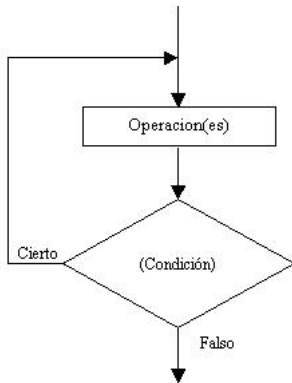


Figura: Diagrama de flujo estructura repetitiva do-while.

```
1 #include <stdio.h>
2 int main()
3 {
4     int numero=0;
5     do
6     {
7         printf("Ingrese un numero entero\n");
8         printf("Para salir ingrese -1\n");
9         scanf("%d",&numero);
10    } while(numero != -1);
11    return (0);
12 }
```

# Estructuras while y do-while: ejemplos I

- 1 Diseñar y codificar un programa que cuente la cantidad de números enteros ingresados por teclado. Cuando el operador ingresa el número -1 se debe dejar de contar e imprimir el resultado.  
[▶ Ver en github](#)
- 2 Diseñar y codificar un programa que imprima todos los números enteros comprendidos entre 0 y N. N debe ser ingresado por el operador. Nota: se debe usar una estructura repetitiva controlada por centinela. [▶ Ver en github](#)
- 3 Modificar el programa anterior, para que imprima todos los números enteros comprendidos entre  $N_1$  y  $N_2$  ambos recibidos por teclado. [▶ Ver en github](#)
- 4 Diseñar y codificar un programa que imprima la suma de todos los números positivos ingresados por teclado. Al ingresar un número negativo, el programa debe salir del bucle e imprimir el valor de la suma. [▶ Ver en github](#)



# Estructuras while y do-while: ejemplos II

- 5 Diseñar y codificar un programa que permita calcular el promedio de notas de un examen parcial de un curso. El programa debe tomar notas hasta que el operador ingrese la calificación -10.

► [Ver en github](#)