



UNITAT 3

ESTRUCTURES REPETITIVES

Programació
CFGS DAW

Autors:

Carlos Cacho y Raquel Torres

Revisat per:

Lionel Tarazon - lionel.tarazon@ceedcv.es

Fco. Javier Valero – franciscojavier.valero@ceedcv.es

José Manuel Martí - josemanuel.marti@ceedcv.es

2021/2022

Llicència



[CC BY-NC-SA 3.0 ES](https://creativecommons.org/licenses/by-nc-sa/3.0/es/) **Reconeixement – No Comercial – Compartir Igual (by-nc-sa)**

No es permet un ús comercial de l'obra original ni de les possibles obres derivades, la distribució de les quals s'ha de fer amb una llicència igual a la que regula l'obra original. Aquesta és una obra derivada de l'obra original de Carlos Cacho i Raquel Torres.

Nomenclatura

Al llarg d'aquest tema s'utilitzaran diferents símbols per a distingir elements importants dins del contingut. Aquests símbols són:



Important



Atenció



Interessant

ÍNDEX

Introducció	4
Estructura Mentre (WHILE)	4
Estructura Fins a (DO-WHILE)	5
Estructura Per a (FOR).	6
Maneres d'acabar un bucle	7
Elements auxiliars	9
Comptadors	9
Acumuladors	9
Interruptors	9
Agraïments	10

UF3. ESTRUCTURES REPETITIVES

1. INTRODUCCIÓ



Les **instruccions repetitives (o bucles)** són aquelles que **permeten variar o alterar la seqüència normal d'execució d'un programa** fent possible que un grup d'operacions (accions) es repetisca un número determinat o indeterminat de vegades, depenent del compliment d'una condició.

Veurem tres tipus: Bucle Mentre (WHILE), Bucle Fer-Fins a (DO-WHILE) i Bucle Per a(FOR)

2. ESTRUCTURA MENTRE (WHILE)



En l'estructura Mentre o “**WHILE**” el bloc d'accions **es repeteix mentre la condició siga certa**, avaluant-se sempre la condició abans d'entrar en el bucle, per això és possible que les accions no s'executen mai.

Pseudocodi	Ordinograma
<pre>Mentre Condició Fer Instrucció 1 Instrucció 2 ... Instrucció N FiMentre</pre>	

3. ESTRUCTURA FINS A (DO-WHILE)



En l'estructura fins a o “**DO-WHILE**”, el bloc d'instruccions **es repeteix mentre que la condició siga certa**, i la condició s'avalua al final del bloc pel que sempre s'executaran almenys una vegada el bloc d'instruccions.

Pseudocodi	Ordinograma
<p>Repetir</p> <p> Instrucció 1</p> <p> Instrucció 2</p> <p> ...</p> <p> Instrucció N</p> <p>Mentre Condició</p>	<pre>graph TD; Entry(()) --> A1[Accion 1]; A1 --> A2[Accion 2]; A2 --> A3[Accion 3]; A3 --> AN[Accion N]; AN --> D{Condicion}; D -- Verdadero --> Entry; D -- Falso --> Exit(());</pre>

4. ESTRUCTURA PER A (FOR)

En l'estructura Per a o "FOR" es coneix per endavant el nombre de vegades que s'executarà el bloc d'instruccions.



El bloc d'accions **es repeteix mentre que la condició siga certa, avaluant-se sempre la condició **abans d'entrar en el bucle****, per això és possible que les accions no s'executen mai.


Aquesta explicació és idèntica a la del bucle WHILE, però un bucle FOR ha de complir les següents característiques:

1. La variable comptador s'inicialitza amb un valor inicial.
2. La condició sempre ha de ser: $\text{variable_comptador} \leq \text{valor_final}$.
3. En cada interacció, la variable comptador s'incrementa en un determinat valor.

Pseudocodi	Ordinograma
<pre>Per a Var_Cont de ValorInicial a ValorFinal amb Increment = n Instrucció 1 Instrucció 2 ... Instrucció N FiPara</pre>	<pre>graph TD Start(()) --> Init[Iniciar contador] Init --> Cond{Condicion} Cond -- Verdadero --> Acciones[Acciones] Acciones --> Inc[Incrementar contador] Inc --> Cond Cond -- Falso --> End(())</pre> <p>Diagrama de flux que representa el bucle FOR. Comença amb un rectangle 'Iniciar contador'. Després, un rectangle de condició 'Condicion'. Si la condició és 'Verdadero', es va al rectangle 'Acciones', després al rectangle 'Incrementar contador', i es torna a la condició. Si la condició és 'Falso', es va directament a la sortida.</p>

5. MANERES D'ACABAR UN BUCLE

Un dels perills que es corren quan s'escriu un bucle és que aquest no acabe mai, la qual cosa es denomina **bucle infinit**, per a no cometre aquest error greu hem de recordar que les condicions dels bucles han de poder canviar dins del bucle, és a dir que si per exemple utilitzem una variable comparada amb una constant, aquesta variable ha de poder canviar de valor dins del bucle.

 Las estructures repetitives han d'incloure un mecanisme perquè aquestes s'acaben.

Alguns dels mètodes més usats per a aqueixa llabor són els següents:

1. Quan sabem el nombre de vegades que es repetirà l'estructura, utilitzarem un comptador.

Per exemple, en un enunciat del tipus: “imprimir la taula del 7”, sabem que el procés va des d'1 a 10, per tant, usarem un comptador.

```
cont = 1
Mentre cont <=10
    Escriure cont * 7
    cont = cont + 1
FiMentre
```

2. Preguntant si volem seguir en el bucle.

Per exemple , en un exercici del tipus “introduir N alumnes i trobar la seua mitjana”, hem de preguntar si volem introduir més alumnes:

```
...
seguir="s"
Mentre ((seguir="s") o (seguir="S"))
    ...
    Escriure "Introduir més alumnes?"
    Llegir seguir
FiMentre
...
```

3. Usant un valor sentinella.

Així, en el cas del següent problema: “Introduir N notes fins a introduir un 10”:

```
...  
Llegir nota  
Mentre (nota <> 10)  
    ...  
    Llegir nota  
FiMentre  
...
```

4. Usant un interruptor que prendrà el valor lògic True o False.

En un exemple com “Repetir unes certes instruccions mentre la condició siga certa”:

```
...  
Mentre (SW = Vertader)  
    ...  
FiMentre  
...
```


6. ELEMENTS AUXILIARS

Els **elements auxiliars** són **variables que realitzen funcions específiques dins d'un programa**, i per la seua gran utilitat, freqüència d'ús i peculiaritats, convé fer un estudi separat d'aquestes.

6.1 Comptadors

Si repetirem una acció un nombre determinat de vegades i aqueixa variable s'incrementara sempre en una quantitat constant, es denomina **comptador**. Seria útil cridar-la alguna cosa així com CONT, CONTA, COMPTADOR... Si tinguérem diversos comptadors dins d'un programa podríem cridar-los CONT1, CONT2...

S'utilitzen en els següents casos:

- Per a comptabilitzar el nombre de vegades que és necessari repetir una acció (variable de control d'un bucle).
- Per a comptar un succés particular sol·licitat per l'enunciat del problema. Un comptador ha d'inicialitzar-se a un valor inicial (normalment a zero) i incrementar-se cada vegada que ocorre un succés.

6.2 Acumuladors

Si per contra, aquest objecte es va **incrementant de manera variable** es denomina **acumulador**. Haurem de cridar-la ACU, ACUM, ACUMULA, ACUMULADOR, SUMA, ... o una altra paraula significativa.

S'utilitza en aquells casos en què es desitja obtindre el total acumulat d'un conjunt de quantitats, sent inicialitzat amb un valor zero.

També a vegades cal obtindre el total acumulat com a producte de diferents quantitats, en aquest cas s'inicialitzarà a un.

Per exemple: *imprimir la suma de N edats.*

6.3 Interruptors

Finalment, tenim unes certes variables que poden **prendre dos valors: cert o fals**. A les variables d'aquest tipus se'ls denomina **interruptors o switches** i la seua funció és que unes certes instruccions s'executen mentre tinga un valor determinat.

S'utilitza per a:

- Recordar que un determinat succés ha ocorregut o no en un punt determinat del programa, i poder així realitzar les decisions oportunes.
- Fer que dues accions diferents s'executen alternativament dins d'un bucle.

Per exemple: *introduir N edats i acabar en introduir un 99.*

7. AGRAÏMENTS

Anotacions actualitzades i adaptades al CEEDCV a partir de la següent documentació:

- [1] Anotacions Programació de José Antonio Díaz-Alejo. IES Camp de Morvedre.