



**Unió Europea**

Fons Social Europeu

*L'FSE inverteix en el teu futur*



**GENERALITAT  
VALENCIANA**



I.E.S. Maria Enriqueta  
Gandia

## UF03. - ESTRUCTURES REPETITIVES (bucles)

**PROGRAMACIÓ  
CFGs DAM**

Autor:

José Manuel Martí Fenollosa

Revisat per:

Àngel Olmos Giner

[a.olmosginer@edu.gva.es](mailto:a.olmosginer@edu.gva.es)

2022/2023

# ESTRUCTURES REPETITIVES

## ÍNDEX DE CONTINGUT



1. Introducció
2. Estructura *WHILE*
3. Estructura *FOR*
4. Estructura *DO-WHILE*
5. Maneres d'acabar un bucle
6. Elements auxiliars
  1. Comptadors
  2. Acumuladors
  3. Interruptors

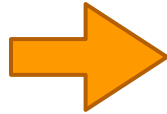
# 1. INTRODUCCIÓ

## INTRODUCCIÓ

### Estructures/Instruccions

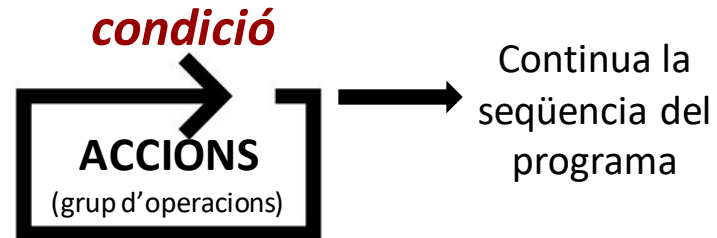
#### Repetitives

(*bucle*)



Permeten **variar o alterar** la **seqüència normal d'execució d'un programa**

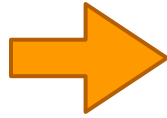
Fan possible que un grup d'operacions (accions) es repetisca **un número determinat o indeterminat de vegades**, depenent del compliment d'una **condició**



# 1. INTRODUCCIÓ

## INTRODUCCIÓ

**3 tipus**



- Bucle WHILE
- Bucle DO-WHILE
- Bucle FOR

# ESTRUCTURES REPETITIVES

## ÍNDEX DE CONTINGUT

1. Introducció
2. Estructura **WHILE**
3. Estructura *FOR*
4. Estructura *DO-WHILE*
5. Maneres d'acabar un bucle
6. Elements auxiliars
  1. Comptadors
  2. Acumuladors
  3. Interruptors

## 2. BUCLE WHILE

### DEFINICIÓ

En l'estructura **WHILE**

Bloc d'instruccions **es repeteix mentre la condició siga certa**

**1er s'avalua la condició** (abans d'entrar en el bucle)

**Es possible que les accions no s'executen mai**

### PSEUDOCODI

*While Condició, Fer*

*Instrucció 1*

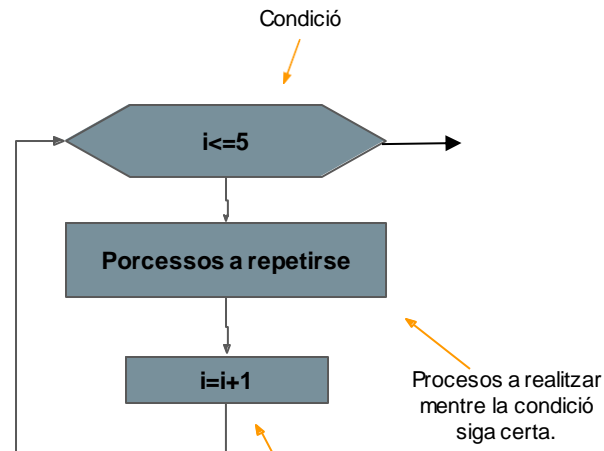
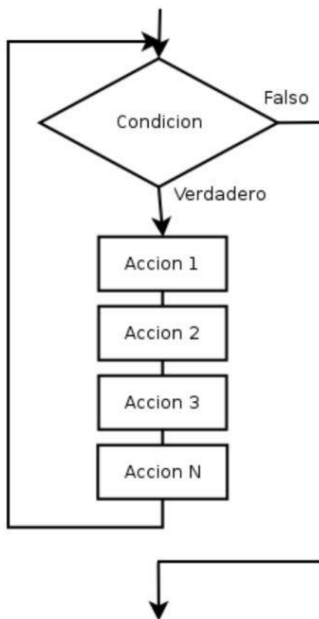
*Instrucció 2*

*...*

*Instrucció*

*N FiWhile*

### ORDINOGRAMA



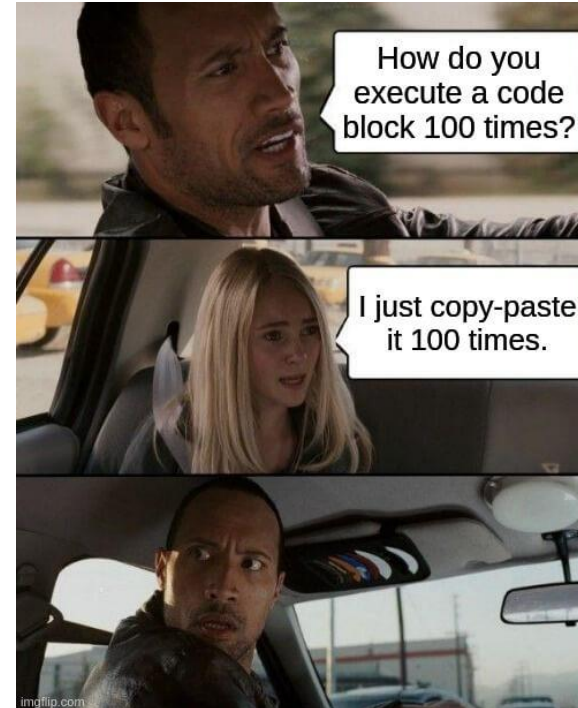
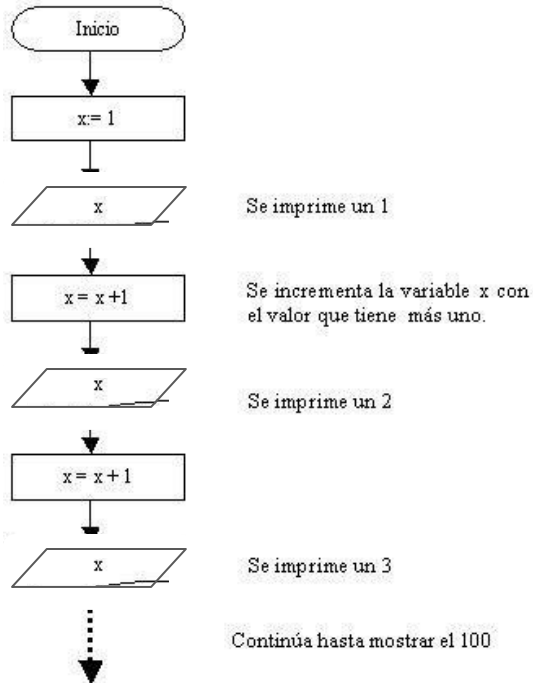
**NO OBLIDEM** fer variar la variable  
DINS del bucle o prodriem generar  
un bucle infinit.

## 2. BUCLE WHILE

### EXEMPLES

**Exemple:** Realitzar un programa que imprimisca en pantalla els números de l'1 al 100

Solució utilitzant  
Estructures Seqüencials

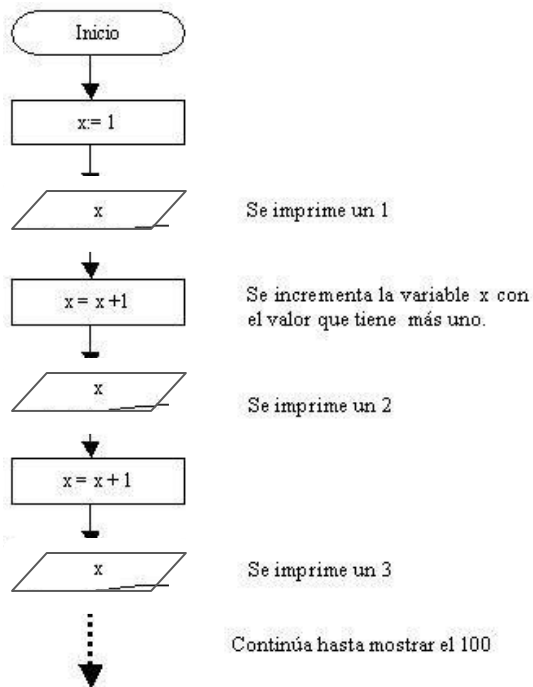


## 2. BUCLE WHILE

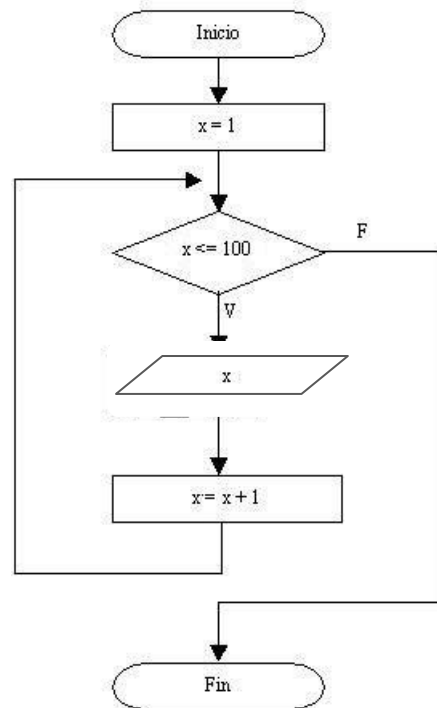
### EXEMPLES

**Exemple:** Realitzar un programa que imprimisca en pantalla els números de l'1 al 100

Solució utilitzant  
Estructures Seqüencials



Solució utilitzant  
Estructures Repetitives





## 2. BUCLE WHILE

### EXEMPLES



GENERALITAT  
VALENCIANA



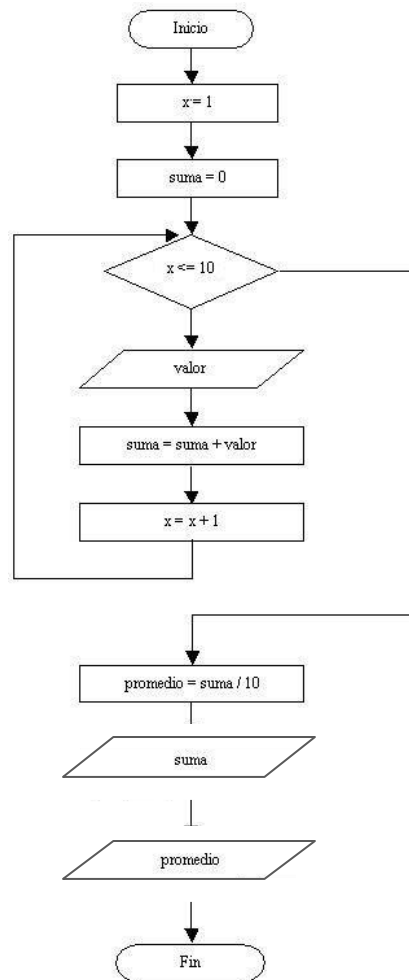
**Exemple:** Desenvolupar un programa que permeta la càrrega de **10 valors per teclat** i ens **mostre** posteriorment **la suma** dels valors ingressats **i la seua mitjana**

COM HO FARÍEU?

## 2. BUCLE WHILE

### EXEMPLES

**Exemple:** Desenvolupar un programa que permeti la càrrega de **10 valors per teclat** i ens **mostre** posteriorment **la suma** dels valors ingressats **i la seua mitjana**



GENERALITAT  
VALENCIANA



U.D. Matemàtiques  
Què és?

# ESTRUCTURES REPETITIVES

## ÍNDEX DE CONTINGUT

1. Introducció
2. Estructura *WHILE*
3. Estructura *FOR*
4. Estructura *DO-WHILE*
5. Maneres d'acabar un bucle
6. Elements auxiliars
  1. Comptadors
  2. Acumuladors
  3. Interruptors

### 3. ESTRUCTURA FOR

#### DEFINICIÓ

En l'estructura **FOR**

Bloc d'instruccions **es repeteix mentre la condició siga certa**

**1er s'avalua la condició** (abans d'entrar en el bucle)

**Es 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:

- La **variable comptador** s'inicialitza amb un **valor inicial**
- La condició sempre ha de ser: **variable\_comptador <= valor\_final**
- En cada interacció, la variable **comptador** es modifica (**++,-- ...**) en un determinat **valor**

### 3. ESTRUCTURA FOR DEFINICIÓ

#### PSEUDOCODI

*FOR Var\_Cont: de ValorInicial  
a ValorFinal amb Increment = n*

*Instrucció 1*

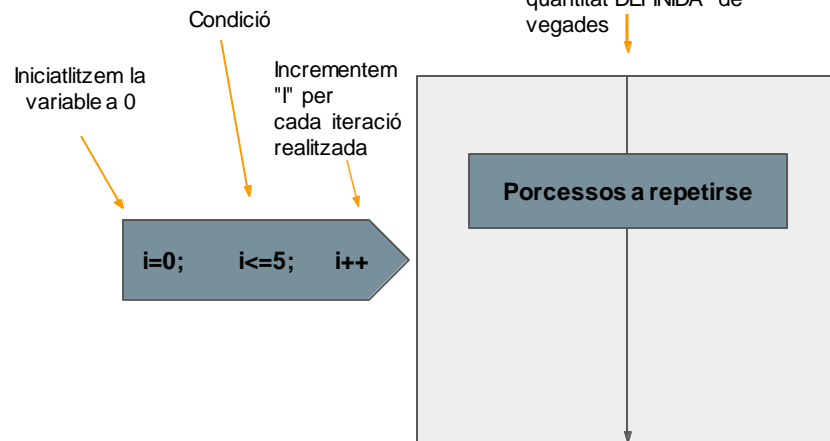
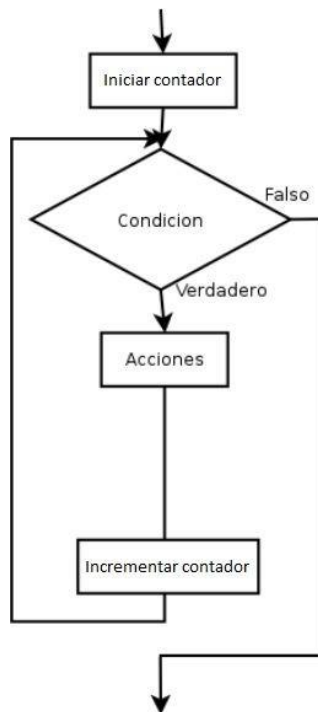
*Instrucció 2*

*...*

*Instrucció N*

*FiFOR*

#### ORDINOGRAMA



#### NOTA

- $i++$  equival a  $i=i+1$
- PERÒ podem incrementar de 2 en 2 o de 0.25 en 0.25, ....:  
 $i=i+2$   
 $i=i+0.25$

### 3. ESTRUCTURA *FOR* *EXAMPLES*

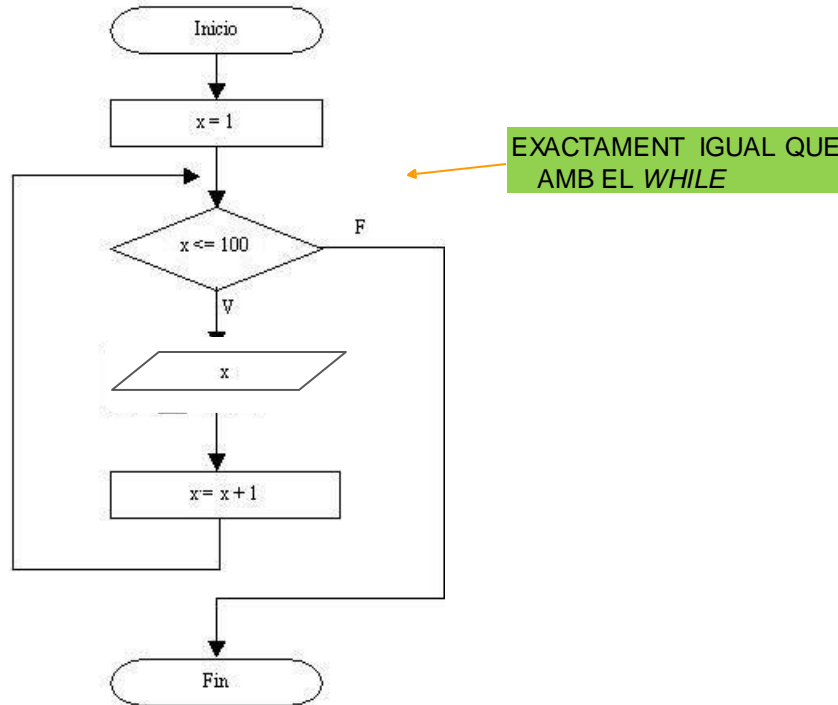


**Exemple:** Realitzar un programa que imprimisca en pantalla els números de l'1 al 100

COM HO FARÍEU?

### 3. ESTRUCTURA FOR EXEMPLES

**Exemple:** Realitzar un programa que imprimisca en pantalla els números de l'1 al 100



### 3. ESTRUCTURA FOR EXEMPLES

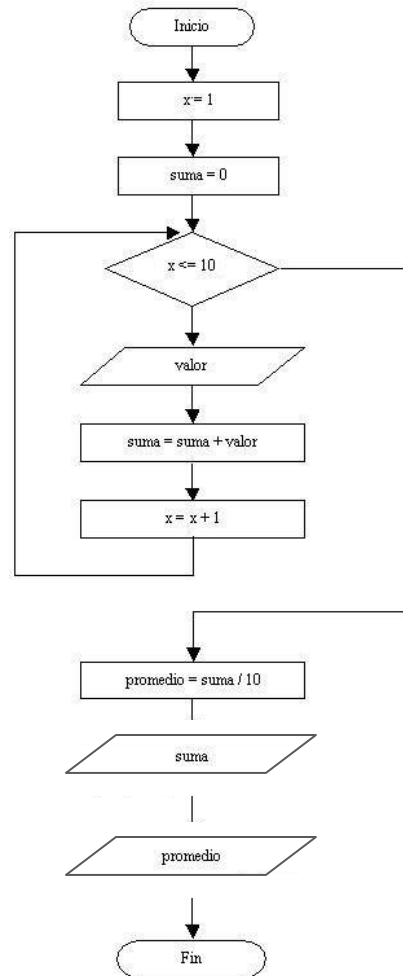
**Exemple:** Desenvolupar un programa que permeti la càrrega de 10 valors per teclat i ens mostre posteriorment la suma dels valors ingressats i la seua mitjana

COM HO FARÍEU?



### 3. ESTRUCTURA FOR EXEMPLES

**Exemple:** Desenvolupar un programa que permeta la càrrega de 10 valors per teclat i ens mostre posteriorment la suma dels valors ingressats i la seua mitjana



EXACTAMENT IGUAL QUE  
AMB EL WHILE



### 3. ESTRUCTURA *FOR* EXEMPLES



GENERALITAT  
VALENCIANA



**Exemple:** *Escriure un programa que llija 10 nombres enters i després mostre quants valors ingressats **eren múltiples de 3 i quants de 5** (no quins). Hem de tindre en compte que hi ha números que són múltiples de 3 i de 5 alhora*

COM HO FARÍEU?

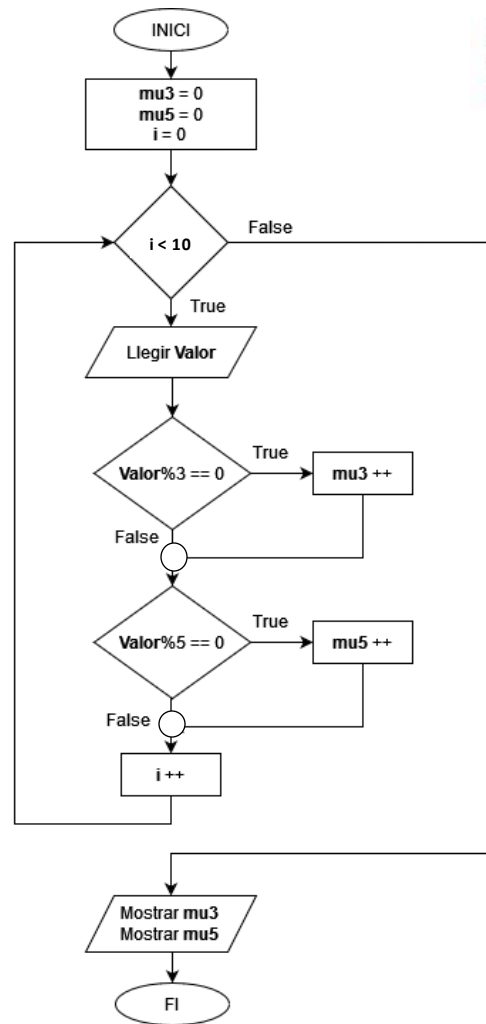
### 3. ESTRUCTURA FOR

#### EXEMPLES

**Exemple:** Escriure un programa que llija **10 nombres enters** i després mostre **quants valors ingressats eren múltiples de 3 i quants de 5** (no quins). Hem de tindre en compte que hi ha números que són múltiples de 3 i de 5 alhora

Per què no hem disposat una estructura IF niada? Perquè hi ha valors que són múltiples de 3 i de 5 alhora. Per tant amb IF niats no podríem analitzar els dos casos o seria més complex de sentenciar.

És important adonar-se quan convé emprar IF niats i quan no ha d'emprar-se.



### 3. ESTRUCTURA *FOR*

*En què es diferencien les estructures WHILE i FOR?*

#### WHILE vs FOR

L'estructura **FOR** repeteix els processos una quantitat **DETERMINADA** de vegades, és a dir, coneixem quantes vegades es repetiran els processos.

Mentre que l'estructura **WHILE** es repetirà una quantitat **INDETERMINADA** de vegades, mentre es complisca la condició.

A l'hora de transformar el diagrama de flux a codi, decidirem quina utilitzar.

# ESTRUCTURES REPETITIVES

## ÍNDEX DE CONTINGUT

1. Introducció
2. Estructura *WHILE*
3. Estructura *FOR*
- 4. Estructura *DO-WHILE***
5. Maneres d'acabar un bucle
6. Elements auxiliars
  1. Comptadors
  2. Acumuladors
  3. Interruptors

# 4. ESTRUCTURA DO-WHILE

## DEFINICIÓ



GENERALITAT  
VALENCIANA



En l'estructura **DO- WHILE**

Bloc d'instruccions **es repeteix mentre la condició siga certa**

**La condició s'avalua al final del bloc** (una volta iniciat el bucle)

**Sempre s'executaran almenys una vegada el bloc d'instruccions**

### PSEUDOCODI

*Repetir*

*Instrucció 1*

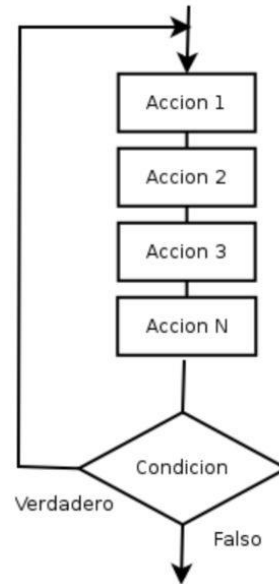
*Instrucció 2*

*...*

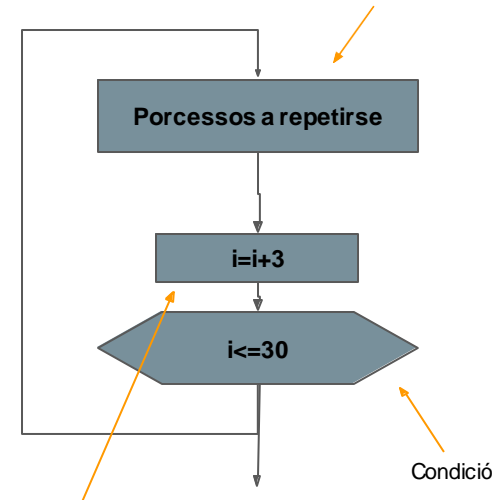
*Instrucció N*

*Mentre Condició*

### ORDINOGRAMA



Procesos a realitzar AL  
MENYS UNA VEGADA i  
mentre la condició siga certa



**NO OBLIDEM** fer variar la variable  
DINS del bucle o podriem generar  
un bucle infinit.

## 4. ESTRUCTURA *DO-WHILE*

### EXEMPLES

**Exemple:** Recupera l'exercici de l'activitat qualificable:

- *de 0 a <3 Molt Deficient.*
- *de 3 a <5 Insuficient.*
- *de 5 a <6 Bé.*
- *de 6 a <9 Notable*
- *de 9 a 10 Excel·lent*

i modifica'l perquè si s'introdueix un número incorrecte,  
no continue i pregunte novament pel número

COM HO FARÍEU?

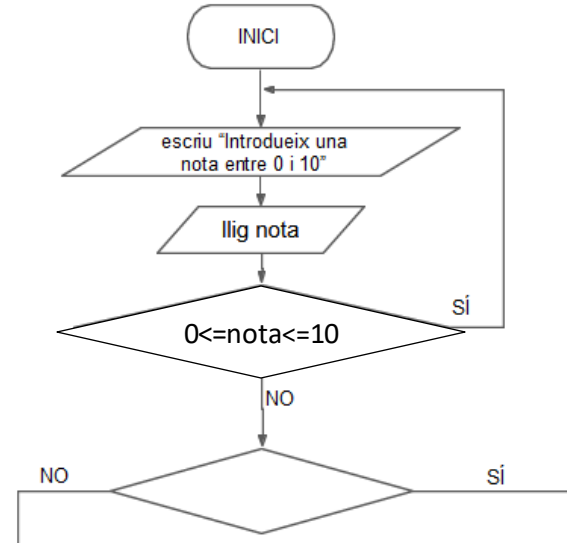
## 4. ESTRUCTURA *DO-WHILE*

### EXEMPLES

**Exemple:** Recupera l'exercici de l'activitat qualificable:

- de 0 a <3 Molt Deficient.
- de 3 a <5 Insuficient.
- de 5 a <6 Bé.
- de 6 a <9 Notable
- de 9 a 10 Excel·lent

i modifica'l perquè si s'introdueix un número incorrecte,  
no continue i pregunte novament pel número





## 4. ESTRUCTURA *DO-WHILE*

### EXEMPLES

**Exemple:** Escriure un programa que sol·licite números entre 0 i 999 (no cal comprovar els negatius), i mostre un missatge de quants dígit té cada número.

Finalitzar el programa quan es carregue el valor 0 (però el 0 també cal analitzar-lo)

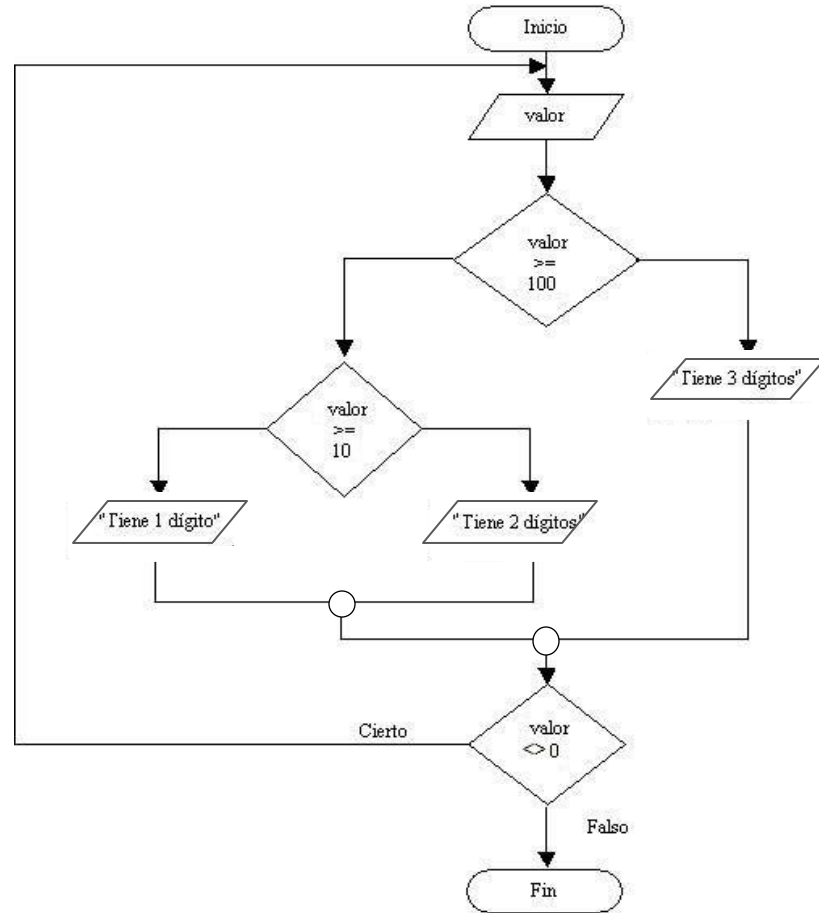
COM HO FARÍEU?

## 4. ESTRUCTURA *DO-WHILE*

### EXEMPLES

**Exemple:** Escriure un programa que sol·licite números entre 0 i 999 (no cal comprovar els negatius), i mostre un missatge de quants dígitos té cada número.

Finalitzar el programa quan es carregue el valor 0 (però el 0 també cal analitzar-lo)



## 4. ESTRUCTURA *DO-WHILE*

### EXEMPLES



GENERALITAT  
VALENCIANA



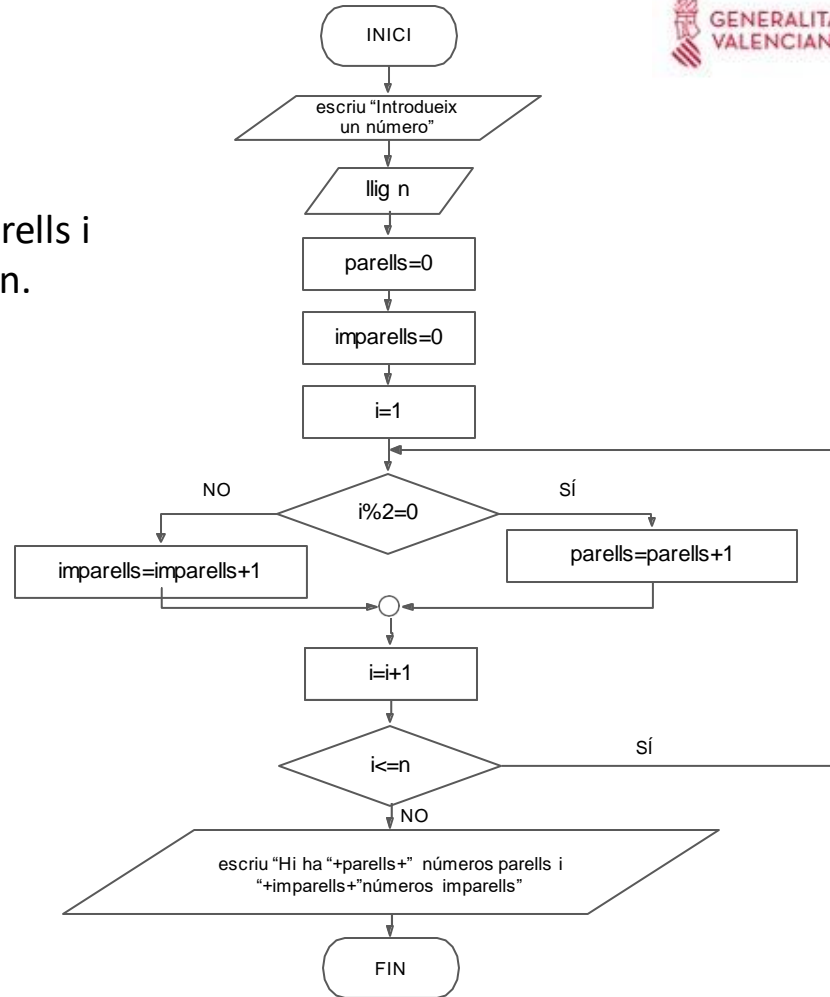
**Exemple:** Dibuixa un ordinograma d'un programa que compte la quantitat de parells i imparells que existeixen en el rang d'1 a n.

COM HO FARÍEU?

## 4. ESTRUCTURA *DO-WHILE*

### EXEMPLES

**Exemple:** Dibuixa un ordinograma d'un programa que compte la quantitat de parells i imparells que existeixen en el rang d'1 a n.



## 4. ESTRUCTURA *DO-WHILE*

*En què es diferencien les estructures WHILE i DO-WHILE?*

QUINA ÉS LA DIFERÈNCIA?

## 4. ESTRUCTURA *DO-WHILE*

*En què es diferencien les estructures WHILE i DO-WHILE?*

### WHILE vs DO-WHILE

*Ambdues es repeteixen mentre es complisca la condició, PERÒ do-while executa almenys una vegada els processos dins d'ell, mentre que un while o un for podrien NO executar els processos*

# ESTRUCTURES REPETITIVES

## ÍNDEX DE CONTINGUT

1. Introducció
2. Estructura *WHILE*
3. Estructura *FOR*
4. Estructura *DO-WHILE*
5. **Maneres d'acabar un bucle**
6. Elements auxiliars
  1. Comptadors
  2. Acumuladors
  3. Interruptors

## 5. MANERES D'ACABAR UN BUCLE

### DEFINICIÓ

Las estructuras repetitivas han d'incloure un **mecanisme perquè aquestes s'acaben**



**bucle infinit**

**Per a no cometre aquest error, 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



## 5. MANERES D'ACABAR UN BUCLE

### DEFINICIÓ



GENERALITAT  
VALENCIANA



**Mètodes més usats per  
a evitar bucles infinits**

1. **Comptador**
2. Preguntar **si volem seguir en el bucle**
3. Usar un **valor sentinella**
4. Usar **interruptor** que prenga valor lògic **True / False**

## 5. MANERES D'ACABAR UN BUCLE

### 1. Comptador

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

Per exemple: “*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
```

A més a més en aquest exemple  
utilitzem el comptador per a generar el  
resultat (molt habitual en programació)

# 5. MANERES D'ACABAR UN BUCLE

## 2. Preguntar si volem seguir en el bucle

### Preguntant si volem seguir en el bucle

Per exemple: “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

...

## 5. MANERES D'ACABAR UN BUCLE

### 3. Usar un valor sentinella

#### Usant un valor sentinella

Per exemple: "Introduir N notes fins a introduir un 10":

...

Llegir nota

Mentre (nota <> 10)

...

Llegir nota

FiMentre

...



És un valor introduït  
per l'usuari

## 5. MANERES D'ACABAR UN BUCLE

### 4. Usar interruptor que prenga valor lògic True / False



GENERALITAT  
VALENCIANA



Usant un interruptor que prendrà el valor lògic True o False

Per exemple: “Repetir unes certes instruccions mentre la condició siga certa”:

...

Mentre (SW = Vertader)

...

FiMentre

...



No ha de ser  
un valor  
introduït per  
l'usuari

# ESTRUCTURES REPETITIVES

## ÍNDEX DE CONTINGUT



GENERALITAT  
VALENCIANA



1. Introducció
2. Estructura *WHILE*
3. Estructura *FOR*
4. Estructura *DO-WHILE*
5. Maneres d'acabar un bucle
6. **Elements auxiliars**
  1. Comptadors
  2. Acumuladors
  3. Interruptors

## 6. ELEMENTS AUXILIARS

### DEFINICIÓ

Els **elements auxiliars** són **variables** que realitzen funcions específiques dins d'un programa

3 tipus



1. Comptadors
2. Acumuladors
3. Interruptors

## 6. ELEMENTS AUXILIARS

### 1. Comptadors

Si **repetirem una acció** un nombre determinat de vegades i aquesta **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 ocórrega un succés.



# 6. ELEMENTS AUXILIARS

## 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 (SUMA)**, sent **inicialitzat amb un valor 0**.
- També a vegades cal **obtindre el total acumulat com a producte de diferents quantitats (MULTIPLICACIÓ)**, **en aquest cas s'inicialitzarà amb ...**

## 6. ELEMENTS AUXILIARS

### 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 (SUMA)**, sent inicialitzat amb un valor 0.
- També a vegades cal **obtindre el total acumulat com a producte de diferents quantitats (MULTIPLICACIÓ)**, en aquest cas s'inicialitzarà amb 1.

Per exemple: imprimir la suma de N edats.