

# programación I

<clase> 4 = estructuras  
repetitivas </clase>



UNIVERSIDAD  
NACIONAL DE CUYO



FACULTAD DE INGENIERIA  
en acción continua...

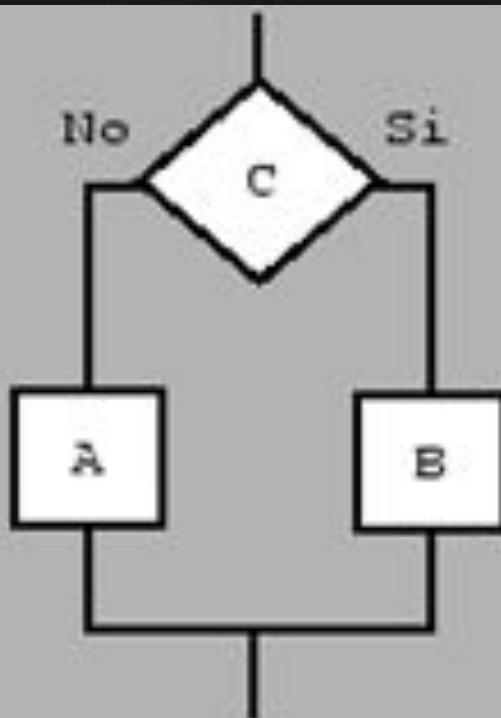
Dra. Elina Pacini  
Lic. Leandro Spadaro  
Ing. Silvina Manganeli  
Lic. Laura Noussan Lettry

# Sentencias de Control

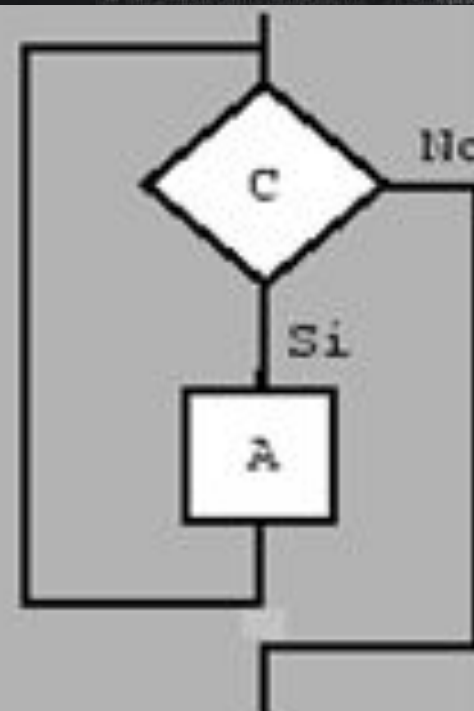
- Permiten controlar o modificar el flujo de ejecución de las instrucciones de un programa



Secuenciales



Condicionales



Repetitivas

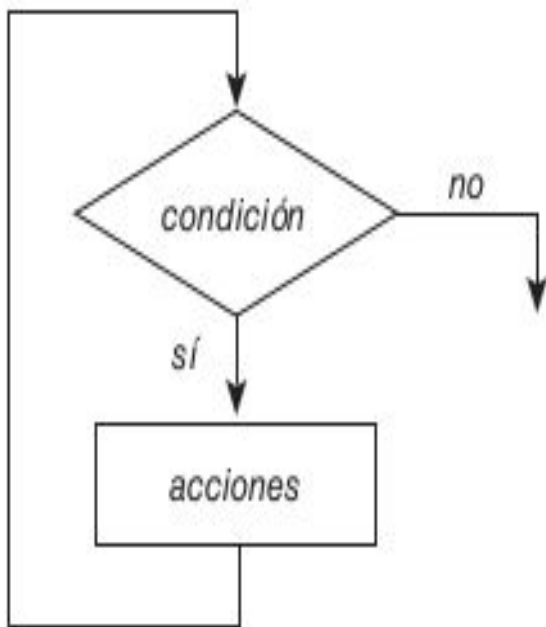
# estructuras repetitivas

- mientras (while)
- hacer-mientras (do-while)
- desde/para (for)

se utilizan cuando se quiere que un conjunto de instrucciones se ejecuten un **número finito** de veces.



# mientras (while)



**Mientras** condicion **Hacer**  
<secuencia\_de\_acciones>  
**Fin Mientras**

condicion: expresion\_logica

# mientras: ejemplo

## Algoritmo cuenta\_enteros\_positivos

Definir contador, num Como Entero

contador = 0

escribir "Ingrese un número entero positivo:"

leer num

**Mientras** num > 0 **Hacer**

    contador = contador + 1

    escribir "Ingrese un número positivo:"

    leer num

**Fin Mientras**

escribir "La cantidad de enteros positivos es ", contador

## Fin Algoritmo



# terminación de bucles con datos de entrada

*Simplemente preguntamos si hay más valores para leer..*

**Algoritmo** suma\_nros

Definir Suma , Num Como entero

Definir resp Como Caracter

Suma = 0

escribir "Desea ingresar números para sumar s/n?"

leer Resp

//la variable Resp es de tipo carácter

**Mientras** Resp = 'S' o Resp = 's' **Hacer**

escribir "ingrese un número entero"

leer Num

Suma = Suma + Num

escribir "Desea sumar más números? (s/n)"

leer Resp

**Fin Mientras**

escribir "La suma de los número ingresados es:", Suma

**FinAlgoritmo**

*Podemos introducir un valor que no cumpla con la condición. Ejemplo:  
Num = 0*

Suma = 0

**leer** Num

**Mientras** Num > 0 **Hacer**

suma = suma + Num

escribir "ingrese un número positivo"

**leer** Num

**Fin Mientras**

*Qué nos mostraría este bloque de código?*

i = 0

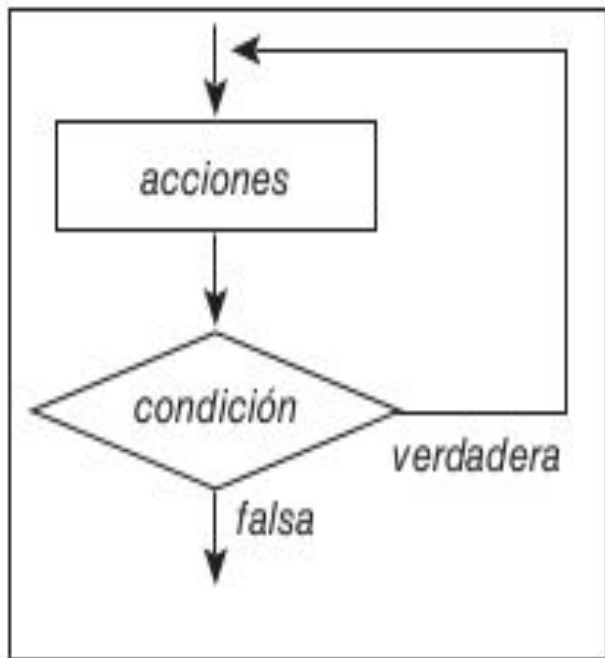
**Mientras** i < 6 **Hacer**

escribir i

i = i + 1

**Fin Mientras**

# hacer-mientras (do-while)



Hacer

<secuencia\_de\_acciones>

Mientras Que

condicion

condicion: expresion\_logica



# hacer-mientras: ejemplo

**Algoritmo** invertir\_numero

Definir num, digitoSig como Entero

num = 198

**escribir** "Número: ", num

**escribir** "Número en orden inverso: "

**Hacer**

digitoSig = num MOD 10

**escribir sin saltar** digitoSig

num = trunc(num / 10)

**Mientras Que** num > 0

**Fin Algoritmo**



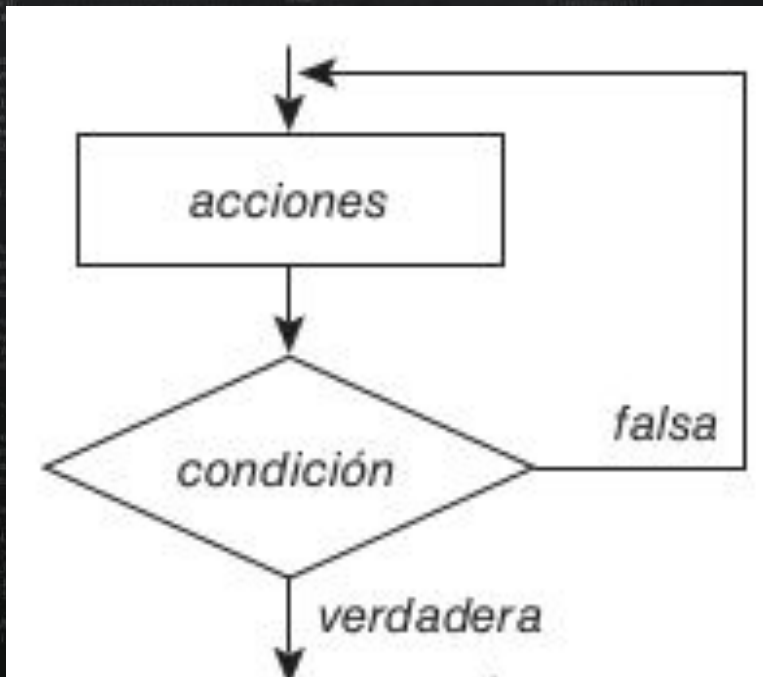
# diferencia entre los bucles mientras y hacer-mientras

```
while (not edge) {  
    run();  
}
```

```
do {  
    run();  
} while (not edge);
```



# repetir (repeat)



Repetir

<acciones>

Hasta Que

<condicion>



**NO UTILIZAMOS ESTE  
BUCLE EN LA  
ASIGNATURA, EN SU  
LUGAR SE UTILIZA:**

**Hacer - Mientras Que**



# repetir: ejemplo

Algoritmo uno\_cien  
Definir num como Entero

num = 1

**Repetir**

escribir num

num = num + 1

**Hasta Que** num > 100

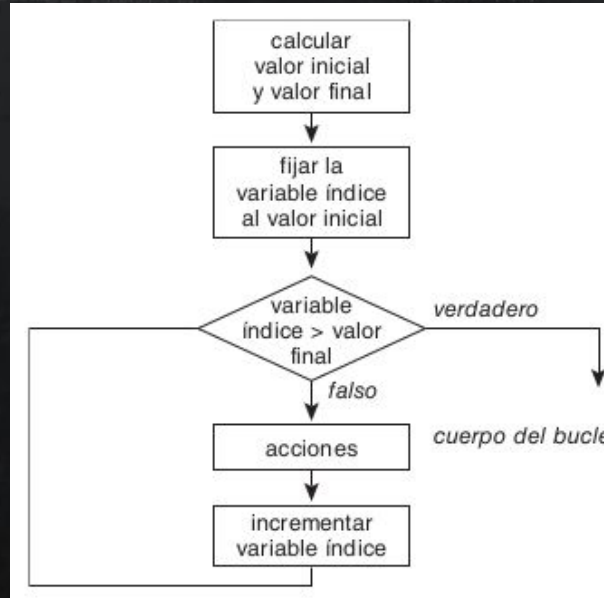
**Fin Algoritmo**



**RECORDAR QUE NO UTILIZAMOS  
ESTE BUCLE EN LA ASIGNATURA**



# para (for)



**Para**  $i \leftarrow vi$  **Hasta**  $vf$  **Con Paso** [incremento/decremento] **Hacer**  
<secuencia\_de\_acciones>

**Fin Para**

$i$ : variable índice numérica

$vi$ : valor inicial de la variable  $i$

$vf$ : valor final a alcanzar por la variable  $i$

# para: ejemplos

## Algoritmo Suma

Definir  $i$ ,  $N$ ,  $S$  como Entero

$S = 0$

Escribir "Ingrese un número entero positivo:"

leer  $N$

Para  $i \leftarrow 1$  Hasta  $N$  Con Paso 1 Hacer

$S \leftarrow S + i$

Fin Para

escribir "La suma de números desde 1 hasta ",  $N$ , " es de ",  $S$

## Fin Algoritmo

Para  $i \leftarrow 20$  Hasta 10 Con Paso 1 Hacer

<acciones>

Fin Para

Para  $i \leftarrow 20$  Hasta 10 Con Paso -1 Hacer

<acciones>

Fin Para





# Selección de bucles

## mientras

- el cuerpo del bucle puede no ser ejecutado
- la repetición no está controlada por contador
- se utiliza cuando se desea omitir el bucle si la condición es falsa

## hacer - mientras

- se usa cuando se debe asegurar que el bucle se ejecute al menos una vez

## para

- se usa cuando el número de repeticiones se conoce por adelantado
- el bucle es controlado por un contador



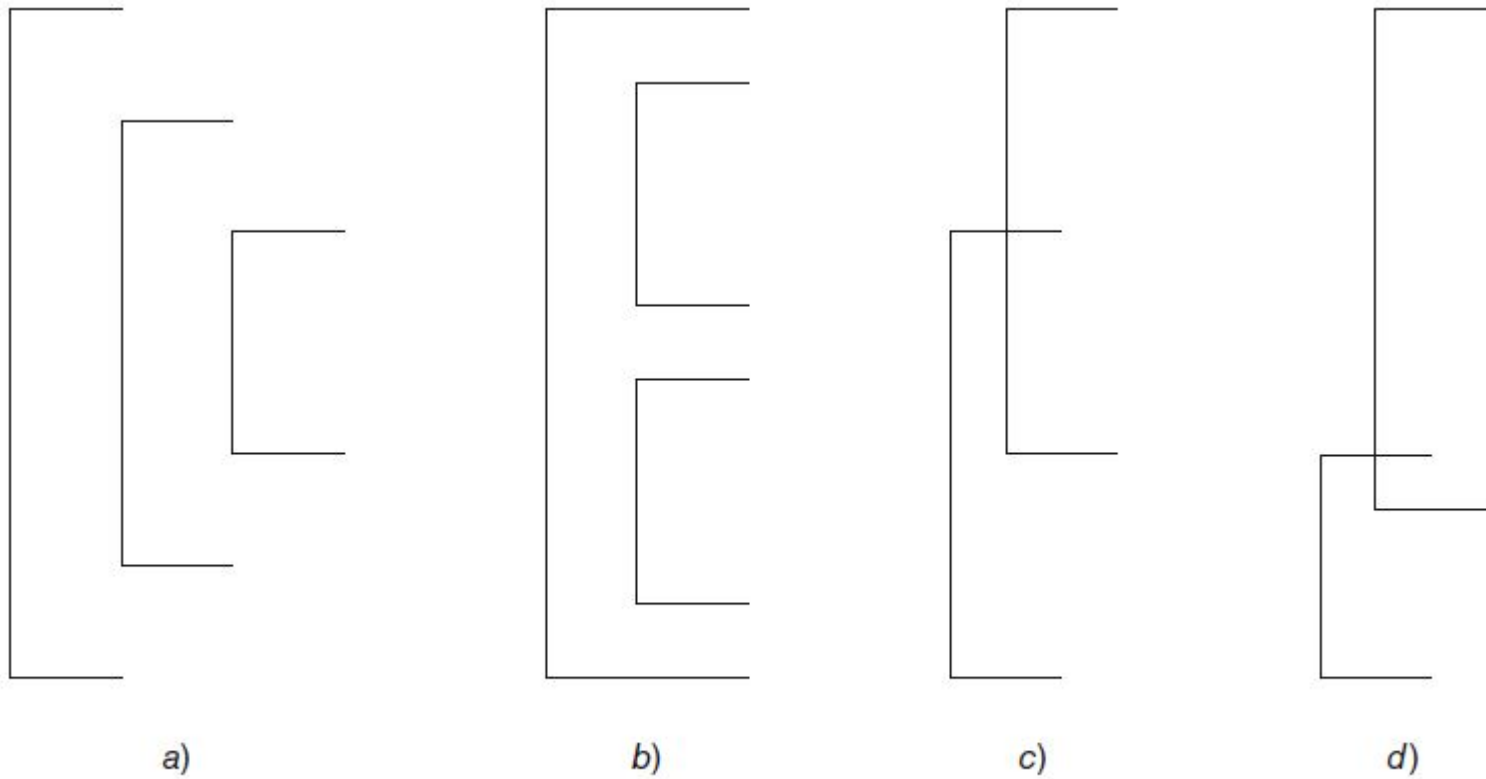
# diseño de bucles

- el cuerpo del bucle
- las sentencias de inicialización de contadores, sumadores, etc.
- la condición para la terminación del bucle

# fin de un bucle

- cuando se conoce el tamaño de la lista
- preguntar antes de la iteración
- valor centinela: valor distinto a todos los posibles de la lista
- agotamiento de la entrada: fin de archivo

# estructuras repetitivas anidadas



Bucles anidados: a) y b), correctos; c) y d), incorrectos.



# estructuras repetitivas anidadas

## Algoritmo MayorPoblacion

Definir i, j, numProv, numCiudades como entero

Definir mayor, poblacion como entero

Escribir "Ingrese número de provincias:"

leer numProv

i = 1

Mientras i <= numProv Hacer

mayor = 0

Escribir "Ingrese numero de ciudades de la provincia ", i

leer numCiudades

j = 1

Mientras j <= numCiudades Hacer

Escribir "Ingrese cantidad de habitantes de la ciudad ", j

leer poblacion

Si poblacion > mayor entonces

mayor = poblacion

Fin Si

j = j + 1

Fin Mientras

escribir "la mayor población de la pcia ", i, "es de: ", mayor

i = i + 1

Fin Mientras

Fin Algoritmo

# estructuras repetitivas anidadas

## Algoritmo numerosFactoriales

Definir i, j, num, n como Entero

Definir factorial como Real

i = 0

Escribir "Ingrese cantidad de números a los cuales desea calcular su factorial : "

leer n //cantidad de números a calcular

**Repetir**

Escribir "Ingrese numero positivo: "

leer num

factorial = 1

**Para** j = 1 **Hasta** num **Con Paso** 1 **Hacer**

factorial = factorial \* j

**Fin Para**

escribir "el factorial del número ", num, " es:", factorial

i = i + 1

**Mientras Que** i < n

**Fin Algoritmo**



# ¿Preguntas?

