

## programación I

<clase> 4 = estructuras
repetitivas </clase>

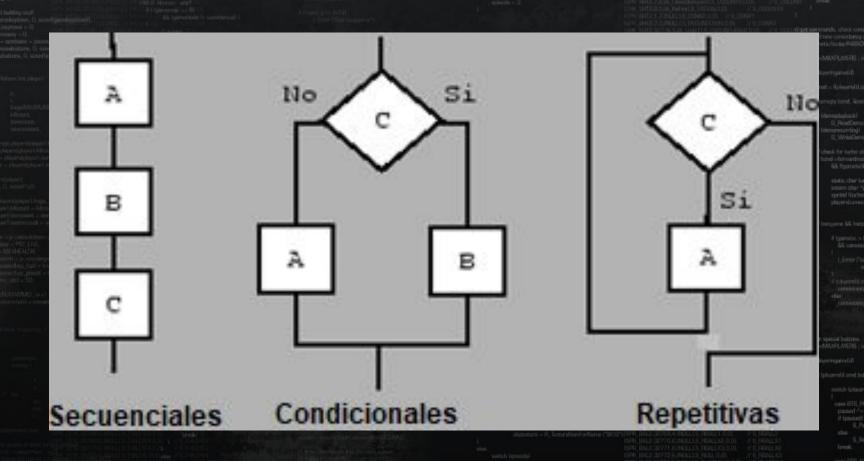




Dra. Elina Pacini Lic. Leandro Spadaro Ing. Silvina Manganelli Lic. Laura Noussan Lettry

### Sentencias de Control

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



### 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.



condición

acciones

SÍ



Mientras condicion Hacer

<secuencia\_de\_acciones>

Fin Mientras

condicion: expresion logica

no

### 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
```

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..

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

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
```

Fin Mientras

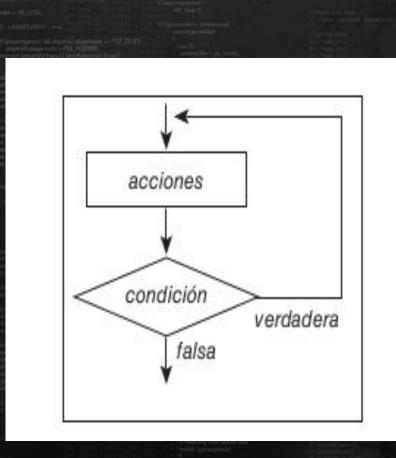
escribir "La suma de los número ingresados es:", Suma

FinAlgoritmo

Qué nos mostraría este bloque de código?

i = 0
Mientras i < 6 Hacer
escribir i
i = i + 1
Fin Mientras</pre>

hacer-mientras (do-while)



Hacer

<secuencia de acciones>

Mientras Que condicion

condicion: expresion logica

### hacer-mientras: ejemplo

```
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
```

Algoritmo invertir nummero

Fin Algoritmo

# diferencia entre los bucles mientras y hacer-mientras



êge in chiarje the game vium processor le gazesthingi ovisch lipitomistikasi

Case (p), hardward G, Drikmerkend ()

C\_Division (p., principation C\_DivisionClaims ) fersile, ==== (p., froi (pario C\_DivisionClaims ) froise.

ireat son un jawahan G. Ballwahana II lasa

C Differential III

Truck

Use gravitory

Start role II

break sare ya worldkine G\_DolAlertsDone II.

use ga, somewhat.

M. Surverdint ()
parvection = pa\_nothing
linesk

.

ind = igenetic ticke #40000PT

for ind ; schill(FLMEHS ; is a)

E interrepretifi

CAMPLAST and a Released Land.

d Manuphyladd G RedDenoTourid Landt

6, And Developed Land), it to was Developed Land)

INTERNAL CONTROL OF THE STATE O

static clar turbornouspattiti colors that "player names 41, spreaf turbornouspe, "No is tur observed to receive the re-

If Indigene SS headons SS (Igeneral

Figuratic > SACKLETICS SS consistent/Sibult to or

Litrar Oscinstanty lakes and scannidary

constructified - phone clar constructified - monte

den ditt

// check for special batters for b=0 ; s-dMAZFL/MERS ; is-1

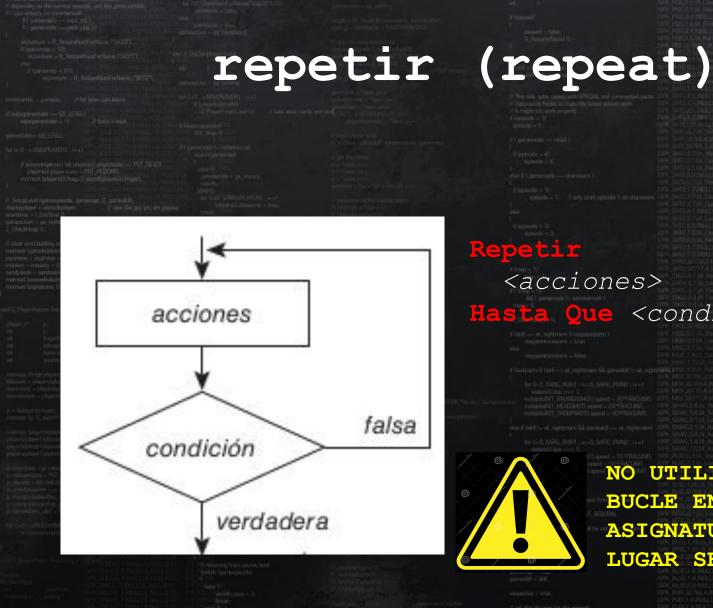
II Iphyeregaine

if lytoperalit and buttons & BT\_9

pased /= 1; d (pased)

S. Prenetioned I elso S. Resurschoos

S\_Reserctions resk:



#### Repetir

#### Hasta Que <condicion>

NO UTILIZAMOS ESTE

BUCLE EN LA

ASIGNATURA, EN SU

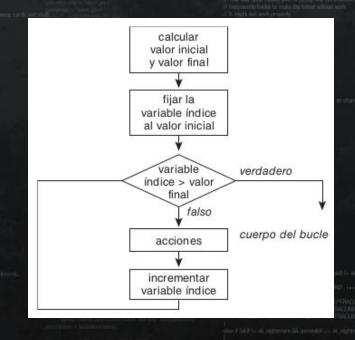
LUGAR SE UTILIZA:

Hacer - Mientras

<acciones>



### para (for)



Para i ← 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

month introduction

consupp, tendinal G DeLeveLand () break

C. Divlandianni II. braid. sees (Jr. Intilizatio G. Divlandianni II. Janes.

Continuedame II formation (i) formation (i) formation (ii) formation (ii) formation (ii) formation (ii) formation (iii)

> trials, use ga\_victory, F\_StartErials (1, locals, user ga\_worldbres,

une gr. joverent.
M. Screethis. ()
generation = gr. jubbr
brook.
seer gr. juding

Wgetpoemands chart canadrae Wgetpijk new consistency chest Suf = ignerationische MINDEPTR

/ physingenelill

(5) cond - Apleyersh Land; (5) memopy land, Saetands()

G Acad Densificant Lin It Literary carrieg! G\_Whit Densificant Is

oden der "player nemes sprief (Uzhammunge, "Na player-Lustioke)ayar Lmis

ll Il Innigame SS hesalemic SS Na

Figuretic > SACKE 65 consistenção

Limit Convenience and con

cher meteoricalitati -

HT HE SIGN for opinion buttoms (I-II) , HAMAZELWERS ; III-II)

2 tokyengametib 1

mitch (playersh)

pased /= 1; d lpined! S.PaseGood

S\_Resurscito treak

in the second

### para: ejemplos

Algoritmo Suma

Definir i, N, S como Entero S = 0

Escribir "Ingrese un número entero positivo:" leer N

Para i←1 Hasta N Con Paso 1 Hacer S ← S + i

Fin Para

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

Para i ← 20 Hasta 10 Con Paso 1 Hacer <acciones>

Fin Para

Para i ← 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

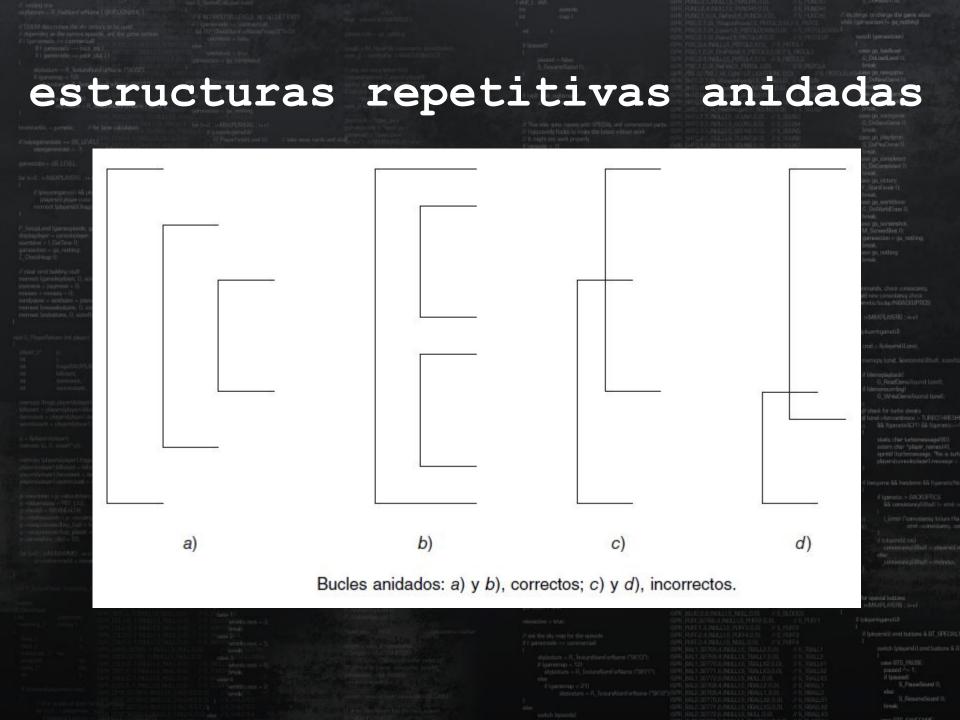
- se usa cuando el número de repeticiones se conoce por adelantado
- el bucle es controlado por un contador

### 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

```
Algoritmo MayorPoblacion
  Definir i, j, numProv, numCiudades como entero
  Definir mayor, poblacion como entero
  Escribir "Ingrese número de provincias:"
  leer numProv
  Mientras i <= numProv Hacer
      mayor = 0
      Escribir "Ingrese numero de ciudades de la provincia ", i
      leer numCiudades
      j = 1
      Mientras j <= numCiudades Hacer</pre>
          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

