

# COMPUTACIÓN (7501)

“A cada día le bastan sus temores, y no hay por qué anticipar los de mañana” C. Péguy

## ESTRUCTURAS ITERATIVAS

### Clase 6

Facultad de INGENIERÍA    Enero-Febrero, 2008

---

---

---

---

---

---

---

---

“A cada día le bastan sus temores, y no hay por qué anticipar los de mañana” C. Péguy

### Agenda

- Revisión
- Estructuras Repetitivas
- For
- While
- Repeat – Until
- Estructuras Anidadas

Facultad de INGENIERÍA    U B A    2008

---

---

---

---

---

---

---

---

“A cada día le bastan sus temores, y no hay por qué anticipar los de mañana” C. Péguy

### Revisión

Todo programa puede ser desarrollado utilizando solamente tres tipos de estructuras de control:

- Secuenciales
- Selectivas
- Repetitivas

Facultad de INGENIERÍA    U B A    2008

---

---

---

---

---

---

---

---

"A cada día le bastan sus temores, y no hay por qué anticipar los de mañana" C. Péguy

## Revisión

Las estructuras selectivas se utilizan para tomar decisiones lógicas; de ahí que se suelen denominar estructuras de decisión o alternativas.

Estructuras Selectivas { If  
Case Switch

Facultad de INGENIERÍA UBA 2008

---

---

---

---

---

---

---

---

"A cada día le bastan sus temores, y no hay por qué anticipar los de mañana" C. Péguy

## Revisión

- Secuenciales - Selectivas

Threads Threads Threads

Facultad de INGENIERÍA UBA 2008

---

---

---

---

---

---

---

---

"A cada día le bastan sus temores, y no hay por qué anticipar los de mañana" C. Péguy

## Revisión

Una condición lógica es una proposición o combinación proposiciones que tiene como respuesta únicamente

Verdadero (V)  
Falso (F)

Facultad de INGENIERÍA UBA 2008

---

---

---

---

---

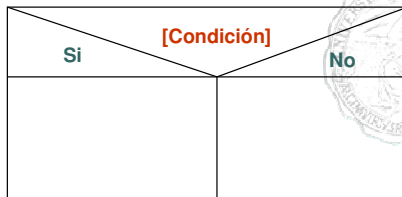
---

---

---

## Revisión

### Diagrama de Nassi-Schneiderman (NS)



## Revisión

### Ejemplos de combinación de condiciones:

Dos o más **Condiciones Lógicas Simples** pueden ser relacionadas por medio de **conectores** formar **Condiciones Compuestas**.

Conectores { or |  
and &



## Estructura If Anidado

Si  $Z_i = 0$

Entonces Muestro  $Z_r$

Sino Si  $Z_r = 0$

Entonces Muestro  $Z_i, i$

Sino Si  $Z_i > 0$

Entonces Muestro  $Z_r, +, Z_i, i$

Sino Muestro  $Z_r, Z_i, i$



## Estructura Case (Pascal)

**Case ... of:** Se utiliza cuando la **variable** es de tipo **ordinal** y puede reemplazar al **if anidado**, es más práctico y eficiente. (Es válida en Pascal)

según...sea E hacer

E1: sentencia 1

E2: sentencia 2

E3: sentencia 3

.....

.....

E<sub>n</sub>: sentencia n

[sino sentencia x]

fin...según



---

---

---

---

---

---

---

---

## Estructura Switch (MatLab)

**Switch:** Se utiliza cuando la **variable** es de tipo **ordinal** y puede reemplazar al **if anidado**, es más práctico y eficiente. (Es válida en MatLab)

según...sea E hacer

E1: sentencia 1

E2: sentencia 2

E3: sentencia 3

.....

.....

E<sub>n</sub>: sentencia n

[sino sentencia x]

fin...según



---

---

---

---

---

---

---

---

## Estructuras Repetitivas

Las estructuras repetitivas o iterativas, se llaman también bucles, lazos o ciclos y se utilizan para realizar varias veces el mismo conjunto de operaciones.



---

---

---

---

---

---

---

---

"A cada día le bastan sus temores, y no hay por qué anticipar los de mañana" C. Péguy

## Estructuras Repetitivas

- Secuenciales - Selectivas - Repetitivas

Facultad de INGENIERÍA UBA 2008

---

---

---

---

---

---

---

---

"A cada día le bastan sus temores, y no hay por qué anticipar los de mañana" C. Péguy

## Estructuras Repetitivas

Las hay de dos tipos:

- donde la cantidad repeticiones se manejan por un número.
- las que se ejecutan hasta que se cumple una condición. (bandera)

Facultad de INGENIERÍA UBA 2008

---

---

---

---

---

---

---

---

"A cada día le bastan sus temores, y no hay por qué anticipar los de mañana" C. Péguy

## Estructuras Repetitivas

Las hay de dos tipos:

	Pascal	MatLab
Número	For	for
Bandera	While	while
	Repeat-Until	-----

Facultad de INGENIERÍA UBA 2008

---

---

---

---

---

---

---

---

## Estructura For

**For:** Esta sentencia, es un bucle controlado por un contador, denominado variable de control o índice.



---

---

---

---

---

---

---

---

## Estructura For

Seudocódigo:

```
para v ← vi hasta vf hacer  
    < acción s1 >  
    < acción s2 >  
    .....  
    < acción sn >  
fin_para
```



---

---

---

---

---

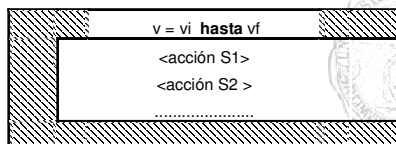
---

---

---

## Estructura For

Diagrama N-S



---

---

---

---

---

---

---

---

## Estructura For (Pascal)

Código:

```
for v := vi to vf do  
  < acción s1 >;
```

```
for v := vi to vf do  
  begin  
    < acción s1 >;  
    < acción s2 >;  
    .....;  
    < acción sn >  
  end;
```

$v_i < v_f$



Facultad de INGENIERÍA

U B A

2008

---

---

---

---

---

---

---

---

## Estructura For (Pascal)

Código:

```
for v := vi downto vf do  
  < acción s1 >;
```

```
begin  
  < acción s1 >;  
  < acción s2 >;  
  .....;  
  < acción sn >  
end;
```

$v_i > v_f$



Facultad de INGENIERÍA

U B A

2008

---

---

---

---

---

---

---

---

## Estructura For (MatLab)

Código:

```
for v = vi : n : vf  
  < acción s1 >;  
end
```

```
for v := vi : n : vf  
  < acción s1 >;  
  < acción s2 >;  
  .....;  
  < acción sn >;  
end;
```

$v_i < v_f$



Facultad de INGENIERÍA

U B A

2008

---

---

---

---

---

---

---

---

## Estructura For (MatLab)

Código:

```
for v = vi : -n : vf  
    < acción s1 >;  
end
```

$V_i > V_f$

```
for v := vi: -n : vf  
    < acción s1 >;  
    < acción s2 >;  
    .....;  
    < acción sn >;  
end;
```



Facultad de INGENIERÍA

U B A

2008

---

---

---

---

---

---

---

---

## Estructura For (Pascal)

Ejemplo 1

Hacer un programa que imprima los números de 5 hasta 28.

```
for R := A to B do { * se define el ciclo A *}  
    writeln ('Numero',R); { * imprime salida * }
```



Facultad de INGENIERÍA

U B A

2008

---

---

---

---

---

---

---

---

## Estructura For (MatLab)

Ejemplo 1

Hacer un programa que imprima los números de 5 hasta 28.

```
for R=A:B % Se define el ciclo  
    fprintf('Numero = %3:0f \n',R);  
end
```



Facultad de INGENIERÍA

U B A

2008

---

---

---

---

---

---

---

---



## Estructura For (Pascal)

### Ejemplo 2

Hacer un programa que calcule e imprima el factorial del número 7



## Estructura For Anidada

Seudocódigo:

```
para v ← vi hasta vf hacer
  para k ← ki hasta kf hacer
    < acción s1 >
    < acción s2 >
    .....
    < acción sn >
  fin_para
fin_para
```



## Estructura For Anidada (Pascal)

Ejemplo 3: Hacer un programa que calcule e imprima el factorial de los números 1 a 7

```
for R := 1 To 7 do { * se define el valor de R *}
begin
  Fac := 1;
  for L := 1 to R do { * se calcula el factorial de R *}
    Fac := Fac * L; { * Calcula factorial *}
  Writeln ('Número : ', R, 'Factorial : ', Fac)
  { * imprime el numero y su factorial *}
end;
```



## Estructura For Anidada (MatLab)

Ejemplo 3: Hacer un programa que calcule e imprima el factorial de los números 1 a 7

```
for R = 1 : 7 { * se define el valor de R *}
    Fac := 1;
    for L = 1 : R { * se calcula el factorial de R *}
        Fac := Fac * L; { * Calcula factorial *}
    end;
    fprintf('Número : %2:0f Factorial : %5:0f\n' R, Fac)
    { * imprime el numero y su factorial *}
end;
```



---

---

---

---

---

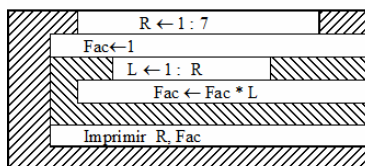
---

---

---

## Estructura For Anidada

Diagrama N-S



---

---

---

---

---

---

---

---

## Bucles Controlados por una Bandera (While – Repeat)

Estos bucles se ejecutan hasta que se cumple una dada condición.

La diferencia entre ambos bucles: **While** y **Repeat**, está en **cuándo** se realiza la pregunta de la condición.



---

---

---

---

---

---

---

---

## Bucle While

En este ciclo la pregunta de condición se hace cada vez **antes de comenzar** el ciclo.

Seudocódigo: **mientras**  $a < b$  **hacer**  
    < acción  $s_1$  >  
    < acción  $s_2$  >  
    .....  
    < acción  $s_n$  >  
**fin\_mientras**



---

---

---

---

---

---

---

---

## Bucle While (Pascal)

Código:

**while**  $a < b$  **do**  
    < acción  $s_1$  >;  
**while**  $a < b$  **do**  
    **begin**  
        < acción  $s_1$  >;  
        < acción  $s_2$  >;  
        .....;  
        < acción  $s_n$  >  
    **end;**



---

---

---

---

---

---

---

---

## Bucle While (MatLab)

Código:

**while**  $a < b$   
    < acción  $s_1$  >;  
**end**  
**while**  $a < b$   
    < acción  $s_1$  >;  
    < acción  $s_2$  >;  
    .....;  
    < acción  $s_n$  >;  
**end;**



---

---

---

---

---

---

---

---

## Bucle While (Pascal)

### Ejemplo 1

Hacer un programa que imprima los números de 5 hasta 28.

```
a := 5;  
while a < 29 do  
begin  
  writeln(a);  
  a := a + 1  
end;
```



---

---

---

---

---

---

---

---

## Bucle While (MatLab)

### Ejemplo 1

Hacer un programa que imprima los números de 5 hasta 28.

```
a=5;  
while a < 29  
  fprintf('%2.0f\n',a)  
  a = a + 1;  
end;
```



---

---

---

---

---

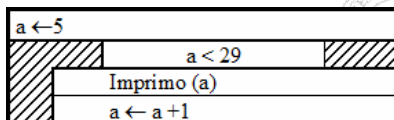
---

---

---

## Bucle While

### Diagrama N-S



---

---

---

---

---

---

---

---

## Bucle Repeat - Until

En este ciclo la pregunta de condición se hace cada vez **al finalizar** el ciclo.

Seudocódigo: **Repetir**

< acción  $s_1$  >

< acción  $s_2$  >

.....

< acción  $s_n$  >

**Hasta** a : v;



---

---

---

---

---

---

---

---

## Bucle Repeat - Until (Pascal)

Código:

**Repeat**

< acción  $s_1$  >;

< acción  $s_2$  >;

.....;

< acción  $s_n$  >

**Until** a < b ;



---

---

---

---

---

---

---

---

## Bucle Repeat - Until (Pascal)

Ejemplo 1

Hacer un programa que imprima los números de 5 hasta 28.

a := 5;

**Repeat**

writeln(a);

a := a + 1

**Until** a < 29;



---

---

---

---

---

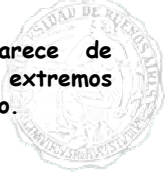
---

---

---

## Bucle Repeat - Until (Pascal)

La estructura **repeat-until** carece de **begin** y **end**, ya que los dos extremos están definidos por el propio ciclo.



---

---

---

---

---

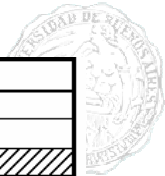
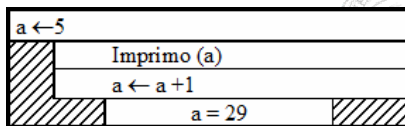
---

---

---

## Bucle Repeat - Until (Pascal)

Diagrama N-S



---

---

---

---

---

---

---

---

## Anidación

**Anidar** significa colocar un ciclo dentro de otro.

Esto quiere decir que una estructura contiene en su interior a otra.

Las estructuras pueden ser de igual o de distinto tipo, siempre que se cumpla con las condiciones siguientes:



---

---

---

---

---

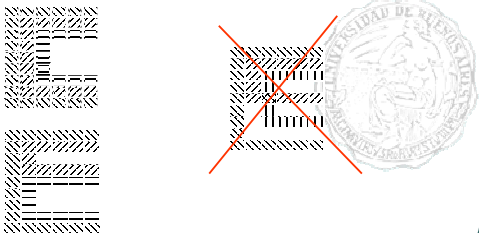
---

---

---

"A cada día le bastan sus temores, y no hay por qué anticipar los de mañana" C. Péguy

## Anidación



M  
g  
i  
n  
g  
F  
e  
r  
n  
a  
n  
d  
o  
J  
L  
A  
G  
E

Facultad de INGENIERÍA U B A 2008

---

---

---

---

---


---

---

---

"A cada día le bastan sus temores, y no hay por qué anticipar los de mañana" C. Péguy

## Preguntas y Respuestas



M  
g  
i  
n  
g  
F  
e  
r  
n  
a  
n  
d  
o  
J  
L  
A  
G  
E

Facultad de INGENIERÍA U B A 2008

---

---

---

---

---

---

---

---

"A cada día le bastan sus temores, y no hay por qué anticipar los de mañana" C. Péguy

## FIN



M  
g  
i  
n  
g  
F  
e  
r  
n  
a  
n  
d  
o  
J  
L  
A  
G  
E

Facultad de INGENIERÍA U B A 2008

---

---

---

---

---

---

---

---