

Resolución de Problemas y Algoritmos

Clase 5

Condicionales “anidados” Repetición incondicional



Dr. Alejandro J. García
http://cs.uns.edu.ar/~ajg



Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Bahía Blanca - Argentina

Palabras Reservadas en Pascal

Listado de las palabras reservadas de Pascal (las que serán utilizadas en RPA están **resaltadas**)

and	end	nil	set
array	file	not	then
begin	for	of	to
case	function	or	type
const	goto	packed	until
div	if	procedure	var
do	in	program	while
downto	label	record	with
else	mod	repeat	

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 2

Condicionales “anidados”

IF <exp. boolean>
THEN

Sentencia (simple o compuesta)

ELSE

Sentencia (simple o compuesta)

IF <exp. boolean E1 >
THEN

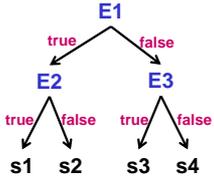
IF < exp. boolean E2 >
THEN <sentencia s. o c. s1>
ELSE <sentencia s. o c. s2>

ELSE

IF < exp. boolean E3 >
THEN <sentencia s. o c. s3>
ELSE <sentencia s. o c. s4>

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 3

Condicionales “anidados”



IF <exp. boolean E1 >
THEN

IF < exp. boolean E2 >
THEN <sentencia s. o c. s1>
ELSE <sentencia s. o c. s2>

ELSE

IF < exp. boolean E3 >
THEN <sentencia s. o c. s3>
ELSE <sentencia s. o c. s4>

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 4

¿Tienen el mismo efecto?

Realice una traza con (i) A=20 y B=0; luego con (ii) A=1 y B=0.

IF (A > 10)
THEN write(1);
IF (B = 0)
THEN write(2);

IF(A > 10)
THEN BEGIN
 write(1);
 IF (B = 0)
 THEN write(2);
END;

¿Por qué con A=1 y B=0 tienen diferente efecto?

En el recuadro de la izquierda (celestes) hay una secuencia de dos sentencias condicionales (if-then) que son independientes entre sí (observe que están separadas por un “;”).

En cambio, a la derecha, como hay un begin-end, el segundo if-then depende del primero ya que está “anidado” dentro del primero: se ejecutará solamente cuando el valor de A sea mayor a 10.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 5

¿Tienen el mismo efecto?

Secuencia de condicionales

IF (A > 10)
 THEN write(1);
IF (B = 0)
 THEN write(2);
IF (C > 20)
 THEN write(3);

Condicionales ANIDADOS:

IF(A > 10)
 THEN write(1)
 ELSE IF (B = 0)
 THEN write(2)
 ELSE IF (C > 20)
 THEN write(3);

Realice diferentes trazas con: A = 20, B = 0, C = 100
A = 1, B = 0, C = 100
A = 1, B = 0, C = 1

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 6

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 “Resolución de Problemas y Algoritmos. Notas de Clase”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2014.

¿Tienen el mismo efecto?

Realice una traza con A=5 y B=6

```

IF A = B
THEN
  IF A = 5
  THEN WRITE('A es 5 ')
ELSE WRITE('DISTINTOS');
```

```

IF A = B
THEN
  BEGIN
    IF A = 5
    THEN WRITE('A es 5 ')
  END
ELSE WRITE('DISTINTOS');
```

El "ELSE" siempre se corresponde con el "IF-THEN" anterior más cercano que no tenga ELSE. Por lo tanto, en el ejemplo de la izquierda el "ELSE" se corresponde con el "IF A=5 THEN". Sin embargo, utilizando "BEGIN - END" puedo forzar y hacer que se corresponda con otro IF-THEN. Esto ocurre en el ejemplo del bloque de la derecha donde el "ELSE" se corresponde con el "IF A=B THEN".

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 7

Problema: Escribir un programa que dado un mes y un año, muestre cuantos días tiene ese mes.

Solución: "30 días trae noviembre, con abril, junio y septiembre, de 28 sólo hay uno, y los demás de 31" pero... en algunos años febrero tiene 29 días, y estos años se llaman años bisiestos ¿cuáles son años bisiestos? ¿por qué pasa esto?

- Un año "astronómico" tiene 365 días 5 h 48 m 45,9 s
- Un año calendario tiene 365 o 366 días (año bisiesto) vea http://es.wikipedia.org/wiki/Año_bisiesto

Definición: un año es **bisiesto** si es múltiplo de 4 y no es múltiplo de 100 o es múltiplo de 400.

Ej. 2004, 2008 y 2000 son bisiestos, 2009, 2010 y 1900 no lo son.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 8

Calcular cuando un año es bisiesto

Definición: un año es **bisiesto** si es múltiplo de 4 y no es múltiplo de 100 o es múltiplo de 400.

Ej. 2004, 2008 y 2000 son bisiestos, 2009, 2010 y 1900 no lo son.

Con una expresión:

```

VAR año:integer; bisiesto: boolean;
bisiesto := (año mod 4=0) and not (año mod 100=0) or (año mod 400=0);
```

Con condicionales:

```

IF año mod 4 = 0
THEN IF año mod 100 = 0
  THEN IF año mod 400 = 0
    THEN bisiesto := true
    ELSE bisiesto := false
  ELSE bisiesto := true
ELSE bisiesto := false
```

Realice trazas con ejemplos significativos

4
100
400
300
2014

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 9

Problema: Escribir un programa que dado un mes y un año, muestre cuantos días tiene ese mes.

Solución: "30 días trae noviembre, con abril, junio y septiembre; 28 o 29 solo hay uno, y los demás de 31" Bisiesto si es múltiplo de 4 y no es múltiplo de 100 o es múltiplo de 400.

Algoritmo

Obtener los valores de mes (1 a 12) y año

Si el mes es 2 (febrero)

entonces: si año es **bisiesto** son 29 días, de lo contrario 28 de lo contrario: (esto es, no es febrero)

si el mes es 11, 4, 6 o 9

entonces son 30 días

de lo contrario son 31

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 10

Posible solución para "días de un mes"

```

PROGRAM CantDiasMes;
VAR mes, año, cant_dias: INTEGER;
BEGIN
write(' Ingrese mes (1 a 12) y año: ');
readln(mes, año);
IF (mes = 2) THEN {... febrero depende si es año bisiesto...}
  IF (año mod 4=0) and (año mod 100<>0) or (año mod 400=0);
  THEN cant_dias := 29
  ELSE cant_dias := 28
ELSE {...en los demás meses depende sólo del mes...}
  IF (mes = 11) OR (mes = 4) OR (mes = 6) OR (mes = 9)
  THEN cant_dias := 30
  ELSE cant_dias := 31;
writeln('La cantidad de días para', mes,' es ', cant_dias);
END.
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 11

Solución con control de ingreso de datos

```

PROGRAM CantDiasMes;
VAR mes, año, cant_dias: INTEGER;
BEGIN {Con control para meses incorrectos}
write(' Ingrese mes (1 a 12) y año: '); readln(mes, año);
IF (mes < 1) OR (mes > 12)
THEN write(' el MES ingresado es incorrecto ')
ELSE BEGIN
  IF (mes = 2) THEN {... febrero depende si es año bisiesto...}
  IF (año mod 4=0) and (año mod 100<>0) or (año mod 400=0)
  THEN cant_dias := 29 ELSE cant_dias := 28
  ELSE {...en los demás meses depende sólo del mes...}
  IF (mes = 11) OR (mes = 4) OR (mes = 6) OR (mes = 9)
  THEN cant_dias := 30 ELSE cant_dias := 31;
  writeln('La cantidad de días para', mes,' es ', cant_dias);
END;
END.
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 12

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 "Resolución de Problemas y Algoritmos. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c)1998-2014.

Otra posible solución para “días de un mes”

```

PROGRAM CantDiasMes;
VAR mes, anio, cant_dias: INTEGER; bisiesto: boolean;
BEGIN
write(' Ingrese mes (1 a 12) y año: '); readln(mes, anio);
bisiesto:= (anio mod 4=0) and (anio mod 100<>0)
           or (anio mod 400=0) ;
IF (mes = 2) AND bisiesto THEN cant_dias := 29;
IF (mes = 2) AND NOT bisiesto THEN cant_dias := 28;
IF (mes = 11) OR (mes = 4) OR (mes = 6) OR (mes = 9)
  THEN cant_dias := 30;
IF (mes=1) or (mes=3) or (mes=5) or (mes=7) or (mes = 8) or
(mes = 10) or (mes=12) THEN cant_dias := 31;
writeln('La cantidad de días para', mes,' es ', cant_dias);
END.
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 13

Otra posible solución para “días de un mes”

```

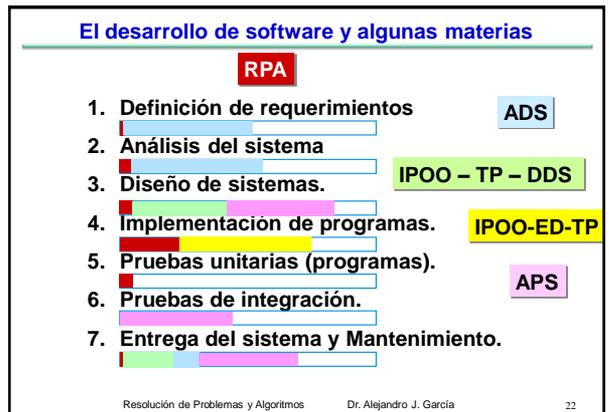
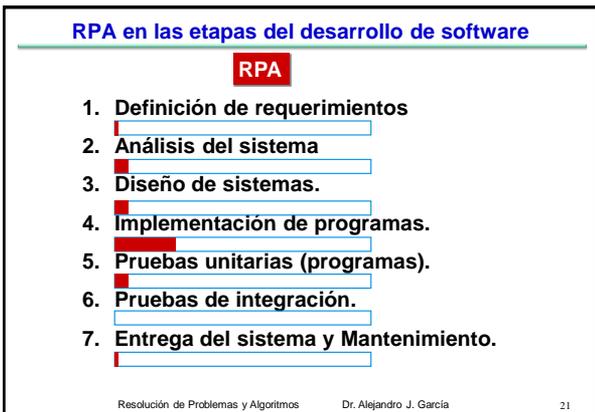
PROGRAM CantDiasMes;
VAR mes, anio, cant_dias: INTEGER;
BEGIN
write(' Ingrese mes (1 a 12) y año: '); readln(mes, anio);
IF (mes = 2) AND (anio mod 4=0) and (anio mod 100<>0)
  or (anio mod 400=0) THEN cant_dias := 29;
IF (mes = 2) AND NOT ((anio mod 4=0) and (anio mod 100<>0)
  or (anio mod 400=0) ) THEN cant_dias := 28;
IF (mes = 11) OR (mes = 4) OR (mes = 6) OR (mes = 9)
  THEN cant_dias := 30;
IF (mes=1) or (mes=3) or (mes=5) or (mes=7) or (mes = 8) or
(mes = 10) or (mes=12) THEN cant_dias := 31;
writeln('La cantidad de días para', mes,' es ', cant_dias);
END.
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 14

•Hacer una traza
•Obs: un único writeln para el resultado



- Etapas en el desarrollo de software**
1. Definición de requerimientos
 2. Análisis del sistema
 3. Diseño de sistemas.
 4. Implementación de programas.
 5. Pruebas unitarias (programas).
 6. Pruebas de integración.
 7. Entrega del sistema y Mantenimiento.
- Ejemplo: páginas web de las materias
- Resolución de Problemas y Algoritmos Dr. Alejandro J. García 20



El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
“Resolución de Problemas y Algoritmos. Notas de Clase”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2014.

Conceptos: Pautas para un buen estilo de programación

- Es sumamente importante que un programa pueda ser **entendido rápidamente** por una **persona** que lo tiene que leer (también los algoritmos).
- Para lograr esto, al escribir programas, hay que tener en cuenta una serie de **pautas** de **“buen estilo de programación”**. Por ejemplo:
 1. Usar tipos de datos adecuados.
 2. Uso de nombres representativos en identificadores.
 3. Indentación (del inglés “indent”).
 4. Comentarios en el código.


```
{...comentarios en Pascal entre llaves...}
// comentarios al finalizar una línea
```

Tipos de datos apropiados

Tipos Predefinidos

- REAL
- BOOLEAN: true, false
- CHAR todos los símbolos del código ASCII
- INTEGER

Tener tipos de datos para las variables permite claridad y abstracción. Dos conceptos fundamentales en el desarrollo, mantenimiento y futuras actualizaciones del software.

Indique un tipo de dato apropiado para cada variable:

```
VAR Precio: REAL;
    Es_Nro_Impar: BOOLEAN;
    Letra: CHAR;
    Día: INTEGER;
    Año: INTEGER;
```

Uso de nombres representativos



```
IF (x1 > x2)
THEN
IF (x1 > w)
THEN z8 := x1
ELSE z8 := w
ELSE IF (x2 > w)
THEN z8 := x2
ELSE z8 := w
```



```
IF (num1 > num2)
THEN
IF (num1 > num3)
THEN máximo := num1
ELSE máximo := num3
ELSE IF (num2 > num3)
THEN máximo := num2
ELSE máximo := num3
```

Indentación



```
IF (num1 > num2)
THEN
IF (num1 > num3)
THEN máximo := num1
ELSE máximo := num3
ELSE IF (num2 > num3)
THEN máximo := num2
ELSE máximo := num3
```



```
IF (num1 > num2)
THEN
    IF (num1 > num3)
        THEN máximo := num1
    ELSE máximo := num3
ELSE
    IF (num2 > num3)
        THEN máximo := num2
    ELSE máximo := num3
```

Comentarios en el código



```
IF (num1 > num2)
THEN
    IF (num1 > num3)
        THEN máximo := num1
    ELSE máximo := num3
ELSE
    IF (num2 > num3)
        THEN máximo := num2
    ELSE máximo := num3
```



```
{ calcula el máximo entre 3
números: num1, num2 y num3}
IF (num1 > num2)
THEN
    IF (num1 > num3)
        THEN máximo := num1
    ELSE máximo := num3
ELSE // num1 <= num2
    IF (num2 > num3)
        THEN máximo := num2
    ELSE máximo := num3
{... la variable máximo ahora tiene
el mayor valor de los 3....}
```

Ahora compare



```
IF (x1 > x2)
THEN
IF (x1 > w)
THEN z8 := x1
ELSE z8 := w
ELSE IF (x2 > w)
THEN z8 := x2
ELSE z8 := w
```



```
{ calcula el máximo entre 3
números: num1, num2 y num3}
IF (num1 > num2)
THEN
    IF (num1 > num3)
        THEN máximo := num1
    ELSE máximo := num3
ELSE // num1 <= num2
    IF (num2 > num3)
        THEN máximo := num2
    ELSE máximo := num3
{... la variable máximo ahora tiene
el mayor valor de los 3....}
```

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente: **“Resolución de Problemas y Algoritmos. Notas de Clase”**. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2014.

En papel también puede usar líneas demarcadoras

☹️

```

IF (num1 > num2)
THEN
  IF (num1 > num3)
  THEN máximo := num1
  ELSE máximo := num3
ELSE
  IF (num2 > num3)
  THEN máximo := num2
  ELSE máximo := num3
                
```

😊

```

IF (num1 > num2)
THEN
  IF (num1 > num3)
  THEN máximo := num1
  ELSE máximo := num3
ELSE
  IF (num2 > num3)
  THEN máximo := num2
  ELSE máximo := num3
                
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 29

De la clase 1:

Para especificar un algoritmo es posible usar:

- (1) **Secuencia**
- (2) **Condiciones**
- (3) **Repeticiones:** permiten especificar de una manera abreviada una secuencia repetida de operaciones.

Llenar botella
 Pasar a bidón
 Llenar botella
 Pasar a bidón
 Llenar botella
 Pasar a bidón
 Enfriar bidón

↔

Repetir 3 veces:

Llenar botella
 Pasar a bidón
 Enfriar bidón

Veremos a continuación como programar en Pascal para que una sentencia se repita **un número fijo de veces.**

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 30

Repetición incondicional en Pascal (p)

Las sentencias de un ciclo **FOR-TO** se ejecutan **CERO** o más veces dependiendo de **valor_inicial** y **valor_final**.

```

FOR V:= valor_inicial TO valor_final
DO 1 sentencia simple
o compuesta ;
Otra sentencia siguiente:
                
```

La sentencia se ejecuta un número fijo de veces y. Juego continua en:

- La **variable V** se suele llamar **variable de control**.
- Al comenzar a **V** se le asigna **valor_inicial**.
- Luego, **V** es **incrementada automáticamente de a uno** en cada repetición (hasta llegar a **valor_final**).

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 31

Ejemplo (p)

Escriba un programa para mostrar por pantalla todos los números enteros entre dos topes ingresados.

```

PROGRAM ListaNumeros;
{Muestra todos los números enteros desde tope inferior a tope superior}
VAR topeinf,topesup,num:INTEGER;
BEGIN
  writeln('Ingrese los topes: ');
  readln(topeinf, topesup);
  writeln('Números');
  FOR num:= topeinf TO topesup
  DO write(num, ' ');
END.
                
```

Ingrese los topes:
 3 8
 Números
 3, 4, 5, 6, 7, 8,

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 32

Ejemplo (p)

Escriba un programa para mostrar por pantalla todos los números enteros entre dos topes ingresados.

```

PROGRAM ListaNumeros;
{Muestra todos los números enteros desde tope inferior a tope superior}
VAR topeinf,topesup,num:INTEGER;
BEGIN
  writeln('Ingrese los topes: ');
  readln(topeinf, topesup);
  writeln('Números');
  FOR num:= topeinf TO topesup -1
  DO write(num, ' ');
  write(topesup)
  {Escribe por separado el último entero para evitar escribir la coma final}
END.
                
```

Ingrese los topes:
 3 8
 Números
 3, 4, 5, 6, 7, 8

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 33

Repetición incondicional en Pascal

```

FOR V:= valor_inicial TO valor_final
DO sentencia
                
```

- **valor_inicial** y **valor_final** son expresiones cuyo valor debe pertenecer al mismo tipo que la variable de control **V**.
- La **sentencia** (que puede ser compuesta), se repetirá un número fijo de veces: **valor_final - valor_inicial + 1**.

FOR V:= 1 TO 5 DO writeln(V);

← repite 5 veces

FOR V:= 5 TO 1 DO write(V);

← repite 0 veces

- Si **valor_final es menor estricto** a **valor_inicial** entonces se repetirá 0 veces.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 34

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 “Resolución de Problemas y Algoritmos. Notas de Clase”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2014.

Repetición incondicional

- Al comenzar a **V** se le asigna el valor inicial y luego, **V** se incrementada automáticamente de a uno hasta llegar al valor final.

<pre>FOR V:= 1 TO 100 DO <sentencia></pre>	Aquí <sentencia> se repite 100 veces
<pre>FOR V:= 100 TO 199 DO <sentencia></pre>	Aquí <sentencia> se repite 100 veces
<pre>FOR V:= -10 TO -1 DO <sentencia></pre>	Aquí <sentencia> se repite 10 veces
<pre>FOR V:= 1 TO -2 DO <sentencia></pre>	Aquí <sentencia> se repite 0 veces

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 35

Escriba un programa para mostrar por pantalla todos los números entre dos topes ingresados en cualquier orden.

```
PROGRAM ListaNumeros;
{muestra los números desde tope inferior a tope superior}
VAR topeinf,topesup,num,aux:INTEGER;
BEGIN
writeln('Ingrese topes: '); readln(topeinf, topesup);
IF topeinf < topeinf {si los topes están invertidos}
THEN begin {intercambio los valores de los topes}
    aux:=topeinf;
    topeinf:=topesup;
    topesup:=aux;
end;
writeln('Números');
FOR num:= topeinf TO topesup
DO writeln(num,', ');
END.
```

Ingrese topes:
8 3
Números
3, 4, 5, 6, 7, 8,

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 36

Sentencia FOR-TO

FOR V:= exp1 TO exp2 DO sentencia

- Tanto **exp1** como **exp2** pueden ser valores, variables, o expresiones siempre que sean del mismo tipo ordinal que **V** (no puede ser tipo real).
- Al comenzar a **V** se le asigna el valor de evaluar **exp1** y luego, **V** se incrementada automáticamente de a uno hasta llegar al valor de **exp2**.

```
N:=2;
FOR V := 1+1 TO N DO writeln(N);
FOR V := N TO N+N DO writeln(N);
FOR V := N div 2 TO (N+10) - N + 2 DO writeln(N);
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 37

FOR-TO con tipo CHAR

```
PROGRAM letras_y_codigos;
{muestra en pantalla los códigos ASCII de algunas letras en columnas}
CONST ultima='J'; columnas=3;
VAR letra: char; contador: integer;
BEGIN
contador:=1;
FOR letra:='A' TO ultima DO
BEGIN {comienzo del ciclo for}
write(letra,'=', ord(letra),' ');
contador:=contador + 1;
{baja de renglón cada "columnas" veces}
if contador mod columnas = 0
then writeln;
END {fin del ciclo FOR}
END.
```

A=65 B=66 C=67
D=68 E=69 F=70
G=71 H=72 I=73
J=74

Comienza con el valor 'A' y luego incrementa automáticamente de a uno pasando por todos los valores del código ASCII hasta llegar al valor final ('J')

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 38

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 “Resolución de Problemas y Algoritmos. Notas de Clase”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2014.