



John Wilder Tukey

Resolución de Problemas y Algoritmos

Clase 5 Condicionales anidados.



Dr. Diego R. García



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

Condicionales IF-THEN-ELSE y IF-THEN

```
IF expresión booleana
THEN Sentencia S1
      (simple o compuesta)
ELSE Sentencia S2
      (simple o compuesta)
...
```

```
IF expresión booleana
THEN Sentencia S1
      (simple o compuesta)
...
```

- Cuando se ejecuta una sentencia “IF”, en primer lugar se evalúa el resultado de la condición (**expresión booleana**) que puede ser **true** o **false**, luego...
- La sentencia S1 que sigue al “THEN” se ejecutará si el resultado de la **expresión booleana** es **true** (si es **false** no se ejecutará).
- En el caso que exista un “ELSE” asociado al “IF”, la sentencia S2 que sigue al “ELSE” se ejecutará únicamente si el resultado de la evaluación de la **expresión booleana** da **false**.

Nuevo problema propuesto

Escriba un programa para hacer un compilador de Pascal.

(mejor lo dejamos para más adelante....)

Vamos algo más simple, que se necesita en un compilador...

Escriba un programa para determinar si una secuencia de caracteres (CHAR) es un identificador válido en Pascal.

(también lo dejamos para más adelante...)

Veamos algo más simple, que se necesita para esto...

Escriba un programa en Pascal que lea un carácter (CHAR) y diga si se trata de una letra mayúscula, minúscula, un dígito o si se trata de otro símbolo.

El código ASCII

American Standard Code for Information Interchange (ASCII)

Está formado por 256 símbolos, aquí se muestran algunos:

				32		33	!	34	"	35	#	36	\$	37	%	38	&	39	'
40	(41)	42	*	43	+	44	,	45	-	46	.	47	/	48	0	49	1
50	2	51	3	52	4	53	5	54	6	55	7	56	8	57	9	58	:	59	;
60	<	61	=	62	>	63	?	64	@	65	A	66	B	67	C	68	D	69	E
70	F	71	G	72	H	73	I	74	J	75	K	76	L	77	M	78	N	79	O
80	P	81	Q	82	R	83	S	84	T	85	U	86	V	87	W	88	X	89	Y
90	Z	91	[92	\	93]	94	^	95	_	96	`	97	a	98	b	99	c
100	d	101	e	102	f	103	g	104	h	105	i	106	j	107	k	108	l	109	m
110	n	111	o	112	p	113	q	114	r	115	s	116	t	117	u	118	v	119	w
120	x	121	y	122	z	123	{	124		125	}	126	~	127		128	Ç	129	ü
130	é	160	á	161	í	162	ó	163	ú	164	ñ	165	Ñ			168	¿		

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”. Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Tipo de dato predefinido de Pascal

Nombre: CHAR (caracter)

Valores: es el conjunto de los 256 caracteres del código ASCII (American Standard Code for Information Interchange)

Operaciones predefinidas: (relacionales) =, >, <, <>, >=, <=

“es posterior en el código ASCII”

“es anterior o igual en el código ASCII”

Para indicar un **valor** de tipo CHAR, se utilizan las comillas simples.
 Ej: 'a', '?', '+', ' ', etc.

Resolución de Problemas y Algoritmos
Dr. Diego R. García
5

Nuevo problema propuesto

Escriba un programa en Pascal que lea un carácter (CHAR) y diga si se trata de una letra mayúscula, minúscula, un dígito o si se trata de otro símbolo.

Por ejemplo:

- 'G', es una mayúscula
- 'g', es una minúscula
- '3', es un dígito
- '\$', es otro símbolo

Siguiendo la metodología propuesta, escriba un algoritmo y un programa en Pascal que resuelva el problema.

Indique cuales son los casos de prueba que usó.

Resolución de Problemas y Algoritmos
Dr. Diego R. 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”. Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Nuevo problema propuesto

Escriba un programa en Pascal que lea un carácter (CHAR) y diga si se trata de una letra mayúscula, minúscula, un dígito o si se trata de otro símbolo.

Algoritmo:

Pedir y leer un carácter ASCII

Si está entre 'A' y 'Z' entonces es una mayúscula

Si está entre 'a' y 'z' entonces es una minúscula

Si está entre '0' y '9' entonces es un dígito

Si no (está entre 'A' y 'Z') y no (está entre 'a' y 'z') y no (está entre '0' y '9') entonces es otro símbolo.

Realice trazas para los casos de prueba: 'G', 'g', '3' y '\$'

Un programa correcto

```

program simbolos;
var ch: char;  mayuscula, minuscula, digito: boolean;
begin {permite distinguir mayúsculas, minúsculas, dígitos y otros símbolos}
write('Ingrese un caracter:');  readln(ch);
mayuscula:= (ch >= 'A') and (ch <= 'Z');
minuscula:= (ch >= 'a') and (ch <= 'z');
digito:= (ch >= '0') and (ch <= '9');
IF mayuscula then writeln(ch, ' es una mayúscula. ');
IF minuscula then writeln(ch, ' es una minúscula. ');
IF digito then writeln(ch, ' es un dígito. ');
IF not mayuscula and not minuscula and not digito
  then writeln(ch, ' es otro símbolo. ');
end.

```

Realice trazas para los casos de prueba: 'G', 'g', '3' y '\$'

Encuentre el error...

```

program ERRONEO; {para ejercitar el uso de casos de prueba}
var ch: char; mayuscula, minuscula, digito: boolean;
begin {¿REALMENTE permite distinguir mayúsculas, minúsculas, dígitos y
      otros símbolos?}
  write('Ingrese un caracter:'); readln(ch);
  mayuscula:= (ch >= 'A') and (ch <= 'Z');
  minuscula:=(ch >= 'a') and (ch <= 'z');
  digito:=(ch >= '0') and (ch <= '9');
  IF mayuscula then writeln(ch, ' es una mayúscula. ');
  IF minuscula then writeln(ch, 'es una minúscula. ');
  IF digito then writeln(ch, ' es un dígito. ')
  ELSE writeln(ch, ' es otro símbolo. ');
end.
    
```



Realice trazas para los casos de prueba: 'G', 'g', '3' y '\$'

¿con cuáles casos de prueba se encuentra el error?

Nuevo problema propuesto

Escriba un programa en Pascal que permita calcular el monto a pagar por un cliente en un comercio. El cliente tiene 3 formas de pago: (E) efectivo (D) tarjeta de débito y (C) tarjeta de crédito. El programa deberá leer el monto y un carácter (E, D o C) para la forma de pago y descontar un 10% si paga con efectivo, o agregar un 5% si paga con crédito.

Indique ejemplos significativos que luego pueda usar como casos de prueba.

Siguiendo la metodología propuesta, escriba un algoritmo y un programa en Pascal que resuelva el problema.

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". Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

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 >
ELSE < sentencia >

ELSE

IF < exp. Boolean E3 >
THEN < sentencia >
ELSE < sentencia >

Resolución de Problemas y Algoritmos Dr. Diego R. García 11

Condicionales “anidados”

```

graph TD
    E1[E1] -- true --> E2[E2]
    E1 -- false --> E3[E3]
    E2 -- true --> s1[s1]
    E2 -- false --> s2[s2]
    E3 -- true --> s3[s3]
    E3 -- false --> s4[s4]
            
```

IF <exp. boolean E1 >
THEN

IF < exp. boolean E2 >
THEN < s1 >
ELSE < s2 >

ELSE

IF < exp. boolean E3 >
THEN < s3 >
ELSE < s4 >

¿Bajo que condiciones se ejecutará la sentencia s3?

¿El resultado de E2 afecta a la ejecución de s3?

Resolución de Problemas y Algoritmos Dr. Diego R. 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”. Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

El límite está en su imaginación

```

graph TD
    E1 -- true --> E2
    E1 -- false --> E3
    E2 -- true --> s1
    E2 -- false --> E4
    E3 -- true --> s3
    E3 -- false --> s4
    E4 -- true --> s5
    E4 -- false --> s6
    
```

¿Bajo que condiciones se ejecutará la sentencia s6?

```

IF <exp. boolean E1 >
THEN
    IF <exp. boolean E2 >
    THEN < s1 >
    ELSE IF < exp. E4 >
    THEN < s5 >
    ELSE < s6 >
ELSE
    IF < exp. boolean E3 >
    THEN < s3 >
    ELSE < s4 >
    
```

Resolución de Problemas y Algoritmos Dr. Diego R. García 13

¿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 los siguientes casos de prueba

- 1) A = 20, B = 0, C = 100
- 2) A = 1, B = 0, C = 100
- 3) A = 1, B = 0, C = 1

¿Qué observa?

Resolución de Problemas y Algoritmos Dr. Diego R. García 14

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”. Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

¿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 (celeste) hay una secuencia de dos sentencias condicionales (**if-then**) que son independientes entre si (observe que están separadas por un “;”).
- En cambio, en el recuadro de la derecha (amarillo), 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.

¿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**”.

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**”. Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Problema propuesto: días de un año

Problema: Escribir un programa que dado un año, indique cuantos días tiene.

Solución:

En general son 365 días pero algunos años febrero tiene 29 días (años bisiestos) y son 366 ¿cuáles son años bisiestos? ¿por qué pasa esto?

- Un año “astronómico” tiene 365 días 5 h 48 m 45,25 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. 2016, 2008 y 2000 son bisiestos, 2009, 2010 y 1900 no lo son.

Problema propuesto: días de un año

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

Con una expresión: (observe que si es mult. de 400 también es de 100 y de 4)

VAR anio:integer; bisiesto: boolean;

bisiesto := (anio mod 4=0) and not (anio mod 100=0) or (anio mod 400=0);

```

program dias_anio;
var anio: integer; bisiesto: boolean;
begin
  write('ingrese año: '); readln(anio);
  bisiesto := (anio mod 4=0) and
    not (anio mod 100=0) or (anio mod 400=0);
  IF bisiesto
    THEN write('tiene 366 días')
    ELSE write('tiene 365 días');
end.
    
```

Casos de prueba:

4
100
400
1900
2000
2014
2015

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 son de 31”, (si el año es bisiesto es 29)

Algoritmo
 Obtener los valores de mes (1 a 12) y año
 Si el mes es 2 (febrero)
 entonces:
 si año es bisiesto
 entonces 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

Casos de prueba:	
mes	año
11	2015
3	2014
12	2014
2	2015
2	2016

Resolución de Problemas y Algoritmos Dr. Diego R. García 19

Posible solución para “días de un mes”

```

PROGRAM CantDiasMes;
VAR mes, anio, cant_dias: INTEGER;
BEGIN {computa la cantidad de días de un mes para un año dado}
write(' Ingrese mes (1 a 12) y año: ');
readln(mes, anio); {no realiza control de ingreso de datos}
IF (mes = 2) THEN {... febrero depende si es año bisiesto...}
    IF (anio mod 4=0) and (anio mod 100<>0) or (anio 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. Diego R. 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”. Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Solución con control de ingreso de datos

```

PROGRAM CantDiasMes;
VAR mes, anio, cant_dias: INTEGER;
BEGIN {computa la cantidad de días de un mes para un año dado}
write(' Ingrese mes (1 a 12) y año: '); readln(mes, anio);
IF (mes < 1) OR (mes > 12) {control de datos ingresados}
THEN write(' el MES ingresado es incorrecto ')
ELSE BEGIN
  IF (mes = 2) THEN {... febrero depende si es año bisiesto...}
  IF (anio mod 4=0) and (anio mod 100<>0) or (anio 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; {del else control de ingreso datos}
END.

```

Resolución de Problemas y Algoritmos

Dr. Diego R. García

21

Otra solución sin anidamiento de IF

```

PROGRAM CantDiasMes;
VAR mes, anio, cant_dias: INTEGER; bisiesto: boolean;
BEGIN {computa la cantidad de días de un mes para un año dado}
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;
IF (mes < 1) OR (mes > 12) {control de datos ingresados}
THEN write(' el MES ingresado es incorrecto ')
ELSE writeln('La cantidad de días para', mes,' es ', cant_dias);
END.

```

Resolución de Problemas y Algoritmos

Dr. Diego R. García

22

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”. Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Expresiones y condicionales anidados

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: (observe que si es mult. de 400 también es de 100 y de 4)
VAR anio:integer; bisiesto: boolean;
 bisiesto := (anio mod 4=0) and not (anio mod 100=0) or (anio mod 400=0);

Casos de prueba:

1900 2000 2014 2015 2016	Una expresión como la de arriba, en la cual se obtienen un resultado para la variable bisiesto, también puede programarse con condicionales anidados:
--------------------------------------	---

➔

Con condicionales:

```

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

Resolución de Problemas y Algoritmos Dr. Diego R. García 23

Expresiones y condicionales anidados

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: (observe que si es mult. de 400 también es de 100 y de 4)
VAR anio:integer; bisiesto: boolean;
 bisiesto := (anio mod 4=0) and not (anio mod 100=0) or (anio mod 400=0);

Casos de prueba:

1900 2000 2014 2015 2016	Una expresión como la de arriba, en la cual se obtienen un resultado para la variable bisiesto, también puede programarse con condicionales anidados:
--------------------------------------	---

➔

Con condicionales:

```

IF anio mod 400 = 0
  THEN bisiesto := true
ELSE IF anio mod 4 = 0
  THEN IF anio mod 100 <> 0
    THEN bisiesto := true
  ELSE bisiesto := false
ELSE bisiesto := false
    
```

Resolución de Problemas y Algoritmos Dr. Diego R. García 24

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”. Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Conceptos: Hardware

Hardware es una palabra inglesa (literalmente: partes duras); como no posee una traducción adecuada, fue admitida por la Real Academia Española que lo define como: *«Conjunto de los componentes que integran la parte material de una computadora»*.

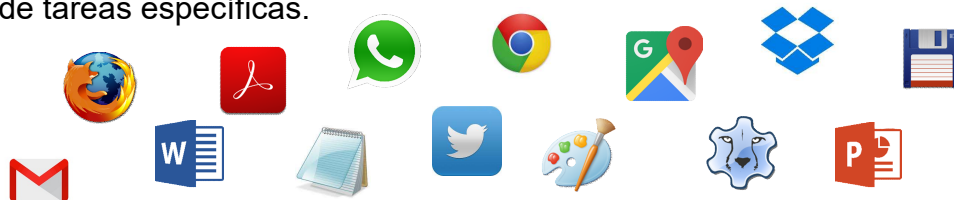
Hardware corresponde a todas las partes tangibles de un sistema informático; sus componentes son: eléctricos, electrónicos, electromecánicos y mecánicos. Son cables, gabinetes, periféricos de todo tipo y cualquier otro elemento físico involucrado; contrariamente, al soporte lógico que es intangible y es llamado **software**.



Conceptos: Software

Software es una palabra inglesa (literalmente: partes blandas o suaves); como en español no posee una traducción adecuada, fue admitida por la Real Academia Española que lo define como: *«Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora »*.

Se conoce como **software** al *equipamiento lógico o soporte lógico* de un sistema informático; comprende el conjunto de los componentes **lógicos** necesarios que hacen posible la realización de tareas específicas.



Software

«Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación»

[Extraído del estándar 729 del [IEEE](#)]

Institute of Electrical and Electronics Engineers (leído i-triple-e)

Considerando esta definición, el concepto de **software** va más allá de los programas de computación; también su documentación, los datos a procesar e incluso la información de usuario forman parte del software: es decir, *abarca todo lo intangible*, todo lo «no físico» relacionado; en contraposición a los componentes físicos, que son llamados **hardware**.

Información adicional

- El término *software* fue acuñado en 1958 por el matemático [John W. Tukey](#) quien usó el concepto de "Software de Computación" (Computer Software) en un artículo publicado en *American Mathematical Monthly*. [2] y [3]
- Además, mientras trabajaba con [John von Neumann](#) en los primeros diseños de las primeras computadoras, Tukey introdujo la palabra "[bit](#)" como acrónimo de las palabras *Binary Digit* ("Dígito binario").



John Wilder Tukey

Para los curiosos:

[1] [SWEBOK](#) (Software Engineering Body of Knowledge)

[2] [SWEBOK 2004.pdf](#)

[3] [SWEBOK 2014.pdf](#)

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*”. Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.