

Unidad 1. Lógica de programación

Tema 2. Tipos básicos de datos y estructuras secuenciales

Tipos básicos de datos y estructuras secuenciales

▶ Contenido

- ▶ Representación y operaciones:
 - ▶ Caracter
 - ▶ Lógicos
- ▶ Operadores de comparación
- ▶ Expresiones:
 - ▶ Lógicas
 - ▶ Mixtas
- ▶ Formatos de E/S en el lenguaje C

▶ Objetivo

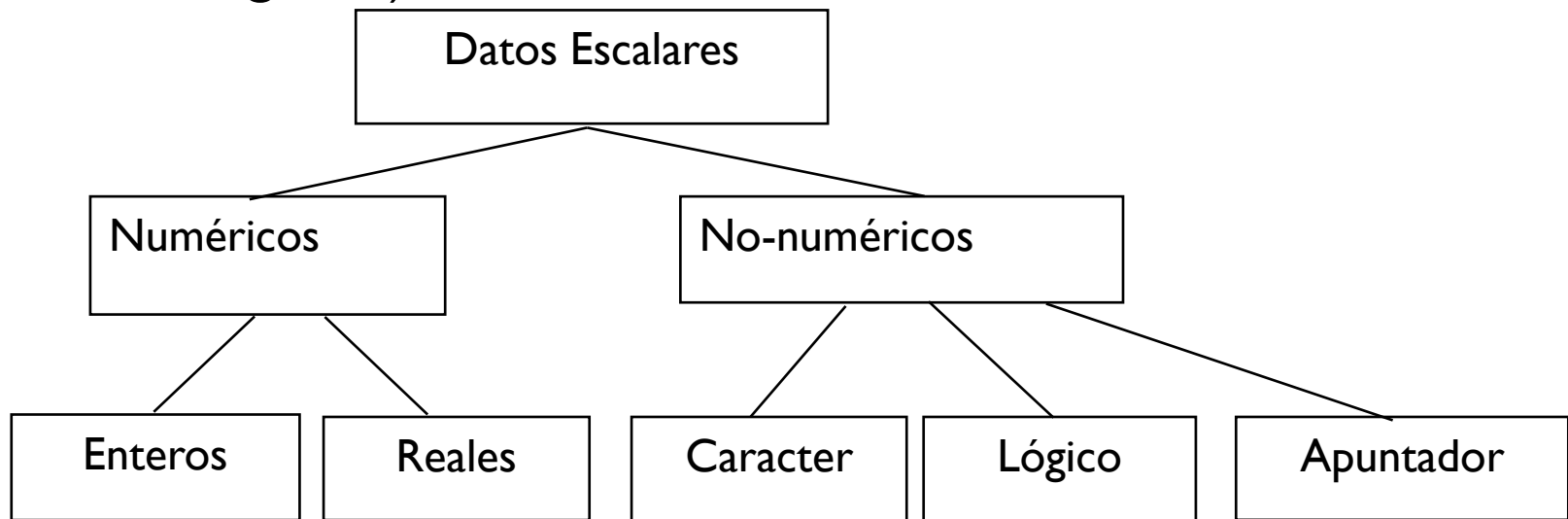
- ▶ Desarrollar habilidades en el uso de los tipos básicos de datos con la visión de TAD y de las estructuras secuenciales

▶ Bibliografía

- ▶ Deitel y Deitel, cap. 2 y 9, sec. 3.11, 3.12, 4.10 y 4.11 y apéndices C y D
- ▶ Navas y Besembel, tema I-III
- ▶ Joyanes, sec. 1.4 - 1.6, 1.9 y 4.4

Tipos de datos

- ▶ Puede clasificarse como escalar o estructurado
- ▶ Escalar o simple: Aquel cuyo dominio presenta una propiedad de orden (Entero, Real, Caracter, Lógico, Apuntador)
- ▶ Estructurado o compuesto: Aquel que se define mediante composición de tipos de datos (vector, cadena de caracteres, matriz, registro)



Tipos de datos escalares en C/C++

Tipo de dato: **Caracter**

C/C++: **char**

- ▶ Conjunto ordenado y finito de los caracteres que la computadora reconoce (letra, dígito, signos de puntuación, etc.)
- ▶ Un caracter es almacenado en un byte usando el código de 8 bits ASCII (American Standard Code for Information Interchange) extendido, lo que da la posibilidad de representar

$2^8 = 256$ caracteres diferentes

Ejemplos

'c' 'e' "' ')" '=' ';' 'D' '7' '@' '<

1100000	1100001	1101101	1100001
---------	---------	---------	---------



¿Qué dice?

ASCII de 7 bits

SYMBOL	ASCII
^	10111110
_	10111111
a	1100001
b	1100010
c	1100011
d	1100000
e	1100101
f	1100110
g	1100111
h	1101100
i	1101001
j	1101010
k	1101011
l	1101000
m	1101101
n	1101110

Tipo de dato Caracter en C/C++

Tipo	Nro. bits
char	8

unsigned char (8) 0 - 256
signed char (8) -127, 127

Operadores:

Solo se leen y escriben

En este lenguaje los caracteres se pueden sumar, restar, multiplicar, dividir, calcular el resto, incrementar y decrementar, sólo si usted los ve como enteros pequeños, si los imprime como caracter obtendrá resultados en ASCII, donde no todos los caracteres son imprimibles

'0064' '006I' '006D' d a m a

Tipos de datos escalares en C/C++

Tipo de dato: Lógico o booleano C/C++: bool

- ▶ Conjunto formado por los valores Verdadero y Falso (TRUE, FALSE)
- ▶ El tipo bool en C++ existe si el compilador lo incluye, pero en C/C++ ANSI no existe, por lo cual el programador lo puede definir mediante la sentencia:

```
enum logico (falso, cierto);
```

- ▶ Operadores:

Negación (\neg)

y lógico (\wedge)

o lógico (\vee)

Tipo de dato Lógico en C/C++

Operación	Operador	Operador en C/C++
Y lógico	\wedge	&&
O lógico	\vee	
No (negación)	\neg	!

Operador unario

!	Resultado
Falso	Cierto
Cierto	Falso

Operadores binarios

&&	Falso	Cierto
Falso	Falso	Falso
Cierto	Falso	Cierto

 	Falso	Cierto
Falso	Falso	Cierto
Cierto	Cierto	Cierto

Expresiones lógicas

- ▶ La que tiene como resultado un valor lógico (cierto o falso)
- ▶ Construcción: mediante los operadores de
- ▶ Comparación ($<$, $>$, $=$, \leq , \geq , \neq)
- ▶ Lógicos (y, o, no)
- ▶ Ejemplos:
 - $40 > -9$
 - $a < b$
 - $z \neq g - 45 + 67 / k$
- ▶ $A = B$ asigna el valor de la variable B a la variable A
- ▶ $A == B$ compara el valor de la variable A con el valor de la variable B
- ▶ Una comparación produce un resultado CIERTO o FALSO

Operadores de comparación

- ▶ Operan sobre los operandos (constantes, variables o expresiones) de diferentes tipos (numéricos o caracteres) y el resultado de su evaluación es un valor lógico (cierto o falso)

- ▶ Ejemplos:

$$(a + b) < c$$

$$a \neq (x*x + b*x + c)$$

$$(\text{horas}*3600) \geq (\text{minutos}*60 + \text{segundos})$$

Operación	Operador	Operador C/C++	Expresión
Igual	=	==	A == B
Diferente	≠	!=	A != B
Menor	<	<	A < B
Menor o igual	≤	<=	A <= B
Mayor	>	>	A > B
Mayor o igual	≥	>=	A >= B

Operadores de comparación en C/C++

A	B	A==B	A!=B	A<B	A<=B	A>B	A>=B
45	67	falso	cierto	cierto	cierto	falso	falso
falso	cierto	falso	cierto				
'p'	'd'	falso	cierto	falso	falso	cierto	cierto
-4.8	-7.99	falso	cierto	falso	falso	cierto	cierto
8	8	cierto	falso	falso	cierto	falso	cierto

!(p && q) || s

p && q && s

(a || b) && (c || d)

a > 5 && !(b < c) || 3 <= d

(op < min) || (op > max)

(edad >= 18) && (edad <= 60)

!encontrado

Jerarquía de los operadores en C/C++

() mayor nivel en la jerarquía

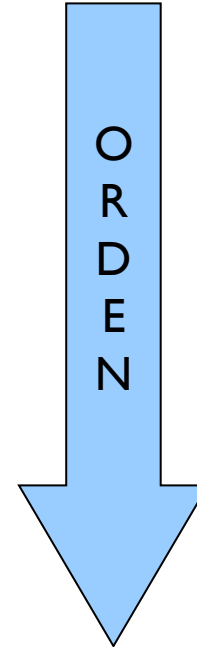
++ -- - (cambio de signo)

* / %

+ - (resta)

< <= > >=

== != menor nivel en la jerarquía



**La evaluación de los operadores con la
misma prioridad se realiza
de izquierda a derecha**

- ▶ Evaluar la siguiente expresión lógica:

$$x \geq (y+z)$$

$$R1 = y + z$$

$$R2 = x \geq R1$$

Para los valores $x = 3, y = 4, z = 1$ se tiene

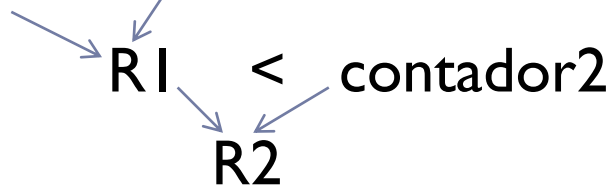
$$R1 = y + z = 4 + 1 = 5$$

$$R2 = x \geq R1 = 3 \geq 5 = \text{falso}$$

Resultado

Evaluar la siguiente expresión lógica:

$$\text{contador1} + 1 < \text{contador2}$$



contador1
contador2

...
3
1
...
...
...

- ▶ Evaluar la siguiente expresión lógica:

$$(a + b - 1) < (x * y)$$

$$R1 = a + b$$

$$R2 = R1 - 1$$

$$R3 = x * y$$

$$R4 = R2 < R3 \quad \Leftarrow \text{Resultado}$$

- ▶ Para los valores $a = 31$, $b = -1$, $x = 3$, $y = 2$

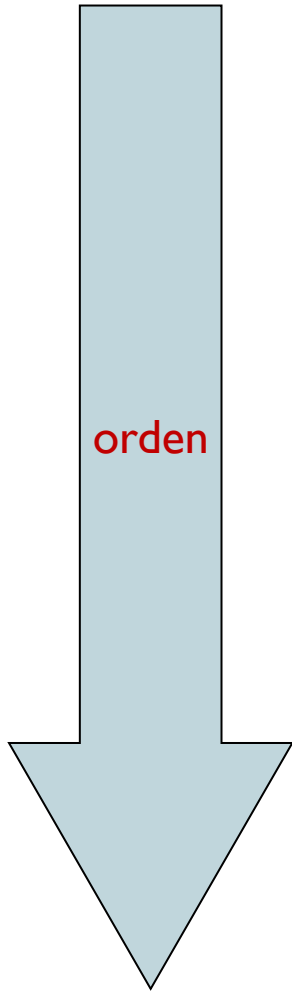
$$R1 = 31 + (-1) = 30$$

$$R2 = 30 - 1 = 29$$

$$R3 = 3 * 2 = 6$$

$$R4 = 29 < 6 = \text{falso} \quad \Leftarrow \text{Resultado}$$

Precedencia y asociatividad de los operadores en C/C++



()

Asociatividad

! ++ -- - (cambio de signo) ← de derecha a izq.

* / %

+ - (resta)

**El resto de
izquierda a
derecha**

< <= > >=

== !=

&&

||

= += -= *= /= %= ← de derecha a izq.

- ▶ Evaluar la siguiente expresión lógica

$$!(x < 5) \ \&\& \ !(y \geq 7)$$

$$R1 = x < 5$$

$$R2 = y \geq 7$$

$$R3 = !R2$$

$$R4 = !R1$$

$$R5 = R4 \ \&\& \ R3 \quad \Leftarrow \quad \text{Resultado}$$

Suponga $x = 3$ $y = 6$

$$R1 = 3 < 5 \quad \rightarrow \quad \text{cierto}$$

$$R2 = 6 \geq 7 \quad \rightarrow \quad \text{falso}$$

$$R3 = !\text{falso} \quad \rightarrow \quad \text{cierto}$$

$$R4 = !\text{cierto} \quad \rightarrow \quad \text{falso}$$

$$R5 = \text{falso} \ \&\& \ \text{cierto} = \text{falso}$$

- ▶ Evaluar la siguiente expresión lógica

$$!((i > 4) \parallel (j \leq 6))$$

$$R1 = i > 4$$

$$R2 = j \leq 6$$

$$R3 = R1 \parallel R2$$

$$R4 = !R3 \quad \Leftarrow \quad \text{Resultado}$$

Suponga $i = 22$ y $j = 3$

$$R1 = 22 > 4 \quad \rightarrow \quad \text{cierto}$$

$$R2 = 3 \leq 6 \quad \rightarrow \quad \text{cierto}$$

$$R3 = \text{cierto} \parallel \text{cierto} \quad \rightarrow \quad \text{cierto}$$

$$R4 = !\text{cierto} \quad \rightarrow \quad \text{falso}$$

- ▶ Evaluar la siguiente expresión lógica

$$!(a + b == c) \parallel (c != 0) \&\& (b - c >= 19)$$

$$R1 = a + b$$

$$R2 = R1 == c$$

$$R3 = c != 0$$

$$R4 = b - c$$

$$R5 = R4 >= 19$$

$$R6 = !R5$$

$$R7 = R3 \&\& R6$$

$$R8 = R7 \parallel R3 \Leftarrow \text{Resultado}$$

Suponga $a = 34$, $b = 12$ y $c = 8$

$$R1 = 34 + 12 = 46$$

$$R2 = 46 == 8 \rightarrow \text{falso}$$

$$R3 = 8 != 0 \rightarrow \text{cierto}$$

$$R4 = 12 - 8 = 4$$

$$R5 = 4 >= 19 \rightarrow \text{falso}$$

$$R6 = !\text{falso} \rightarrow \text{cierto}$$

$$R7 = \text{cierto} \&\& \text{falso} \rightarrow \text{falso}$$

$$R8 = \text{cierto} \parallel \text{falso} \rightarrow \text{cierto}$$

Operadores de asignación en C/C++

Operador	Expresión	Expresión abreviada
=	$a = a + 2$	
+=	$x = x + 5$	$x += 5$
-=	$y = y - z$	$y -= z$
*=	$a = a * b$	$a *= b$
/=	$y = y/2$	$y /= 2$
%=	$mes = mes \% 2$	$mes \% = 2$

La asignación $A += (B * C) + (A * D)$ es válida!!!

Operador coma, sizeof y asignación múltiple

- ▶ Operador coma: Separador de sentencias

```
x = (y = y - 5, 25/y);
```

Si $y=10$, $x=?$
Si $y=5$, $x=?$

- ▶ Operador sizeof: determina el tamaño en bytes de una variable

```
int x;
```

```
printf("%d", sizeof x);
```

¿Qué valor
imprime?

- ▶ Sentencia de asignación múltiple:

$A = B = C = D = 1$ asigna a las variables A, B, C y D el valor de 1

Use asignación múltiple solamente para asignar el **mismo** valor a diferentes variables

Cambio de tipo de dato y formatos para leer en C

- ▶ Operador *cast* u operación *casting*: permite cambiar el tipo de dato declarado para una variable

(tipoDeDato) variable

- ▶ Formatos para leer en C

Formato	Descripción
%c	Leer un caracter
%d	Leer un entero
%i	Leer un entero
%e	Leer un real en punto flotante
%f	Leer un real en punto flotante
%h	Leer un entero corto
%s	Leer una cadena de caracteres

Ejemplo:

`Y=(float)k;`

`scanf("%d %d", &r, &c);`

Un * entre % y la letra, suprime la asignación

`scanf("%*d", &x);`

Un valor entero entre % y la letra, indica el ancho del número de caracteres leídos

`scanf("%10s", &z);`

Formatos de escritura en C

Formato	Descripción
%c	Escribe un caracter
%d	Escribe un entero
%i	Escribe un entero
%e	Escribe un real en notación científica
%f	Escribe un real en punto flotante
%g	Usa %e o %f, el que sea más corto
%u	Escribe un entero sin signo o solo positivos
%s	Escribe una cadena de caracteres
%%	Escribe el caracter %

Si es cadena de caracteres, entero.entero indica el mínimo y el máximo número de caracteres a imprimir
 Si es entero, l o h entre % y la letra, indica que se imprime un long o un short, respectivamente
 Si es real, l entre % y la letra, indica que se imprime un double
`printf(“%li %hd %lg”, w, r, p);`

```
printf(“Hola %c %d %s %f”, s, x, c, y);
```

Un valor entero entre % y la letra, indica el ancho del número a imprimir, un 0 antes indica imprimir ceros antes del valor, un menos antes del entero indica justificar a la izquierda el valor a imprimir y entero.entero indica el ancho.#decimales si es f

```
printf(“%6i %05d %10.4f %5.7s %-10.2f”, s, x, z, y);
```

Diseño	Notación en C/C++
Tipo Entero	short, int, long, long long, unsigned
Tipo Real	float, double, long
Tipo Caracter	char
Tipo Lógico	bool
Constantes	<code>#define IDENTIFICADOR valor</code> <code>const tipoDeDato IDENTIFICADOR=valor;</code>
Declaración de variables	<code>tipoDeDato listaDeVariables;</code>
Lectura : Leer listaDeVariables	<code>scanf("formato", listaDeVariables); /</code> <code>cin >> listaDeVariables;</code>
Escritura: Escribir "comentario", var	<code>printf("comentarioY/Oformato", listaDeVariables); /</code> <code>cout << "comentario" << listaDeVariables << endl;</code>

Ejercicio resuelto

► Enunciado del problema:

Los organizadores de una competencia de atletismo desean tener una traducción inmediata de los resultados de las competencias en unidades del sistema métrico. Conviértase los resultados del salto alto de metros a pies y pulgadas (1 metro = 39,37 pulg, 1 pie = 12 pulg) con precisión de 2 decimales

► Análisis en E-P-S:

Entrada	Variable	Descripción	Tipo de dato	Rango válido
	alturaEnMetros	Variable que contiene el valor de la altura saltada por el atleta en metros	Real	Mayor o igual a cero (≥ 0)
Proceso	$\text{alturaEnPulgadas} = \text{alturaEnMetros} * 39,37$ $\text{alturaEnPies} = \text{alturaEnPulgadas} / 12,00$			
Salida	alturaEnPulgadas	Variable que contiene el valor de la altura saltada por el atleta en pulgadas	Real	≥ 0
	alturaEnPies	Variable que contiene la altura saltada por el atleta en pies	Real	≥ 0



Ejercicio resuelto

Jul. 2010		competencia
	{pre: }	{pos: }
1	alturaEnMetros= valor suministrado	alturaEnMetros: Real+. Valor de la altura saltada en metros
2	alturaEnPulgadas = alturaEnMetros*39,37	alturaEnPies: Real+. Valor de la altura saltada en pies
3	alturaEnPies = alturaEnPulgadas/12	alturaEnPulgadas: Real+. Valor de la altura saltada en pulgadas
4	Desplegar alturaEnPulgadas, alturaEnPies	
1	alturaEnMetros=1.00=>alturaEnPulgadas=39.37, alturaEnPies=3.28	
2	alturaEnMetros=0.00=>alturaEnPulgadas=0.00, alturaEnPies=0.0	

Diseño en TDSO

```
/*      Convierte un valor real en metros a pies
      Isabel Besembel      22/7/2010      */
```

```
#include<stdio.h>
int      main()
{
    float      alturaEnMetros, alturaEnPulgadas, alturaEnPies;
    printf("Por favor ingrese la altura saltada (mts)");
    scanf("%f", &alturaEnMetros);
    alturaEnPulgadas=alturaEnMetros * 39.37;
    alturaEnPies = alturaEnPulgadas/12.00;
    printf("Salto %6.2f (mts) equivalente a %6.2f (pies) y a %6.2f (pulgadas)\n", alturaEnMetros, alturaEnPies, alturaEnPulgadas);
    return 0;
}
```

Codificación en C

- ▶ Cómo se representan los caracteres
- ▶ Cómo se representan los valores lógicos
- ▶ Cuáles son los operadores lógicos
- ▶ Cuáles son los operadores de comparación
- ▶ Cómo son las expresiones lógicas
- ▶Cuál es la jerarquía de los operadores
- ▶ Cuáles son los operadores de asignación, de continuación (coma), de tamaño (sizeof) y cambio de tipo
- ▶ Cómo es una asignación múltiple
- ▶ Cuáles son los formatos de E/S en el lenguaje C

Resumen

¿Cuáles son los conceptos relevantes de esta clase?



- Realice el análisis E-P-S, diseño en TDSO y codificación en C/C++ para cada uno de los enunciados dados a continuación:
1. Calcular el promedio de seis valores con 2 dígitos decimales
 2. Dado el peso de una persona en libras expresado con dos decimales, calcular su peso en kilogramos y gramos expresado con cuatro decimales
 3. Calcular el salario neto (con 2 decimales de precisión) de un trabajador en función del número de horas trabajadas, precio de la hora de trabajo, el sueldo bruto y considerando unos descuentos como: impuestos el 20%, seguro el 4% y ley de política habitacional de 5%
 4. Calcular el valor de $f(x)$ dado el valor de x . Expresar el resultado en notación científica
- $$f(x) = \frac{4}{x^2 + 1}$$



Ejercicios

5. Dado el radio de un círculo (con 4 decimales de precisión), calcular e imprimir su superficie y la longitud de la circunferencia manteniendo la precisión
6. Calcular la suma (S) de los términos de una progresión aritmética que empieza en 2 (a_1), termina en 602 (a_n) y tiene razón 2 (r), sabiendo que:

$$a_n = a_1 + (n - 1).r$$

$$S = \frac{(a_n + a_1)n}{2}$$

7. Una progresión aritmética de n términos empieza en a_1 y la suma de sus términos es S . Calcular la razón (r).
8. El costo del seguro en una localidad agrícola típica es el 3.5% del monto asegurado por hectárea, multiplicado por el número de hectáreas sembradas. Suponiendo que las semillas por cosechar se limitan a maíz, avena y trigo, calcule el costo total de la póliza de seguro dado el monto asegurado y el número de hectáreas sembradas de cada semilla.