

Módulo I: Conceptos Básicos

- Tema 1. ¿Qué es un ordenador?
- Tema 2. ¿Cómo se representan los datos en un ordenador?
- Tema 3. ¿Qué es un lenguaje de programación?
- Tema 4. ¿Cómo se hace un programa informático?
- **Tema 5. Tipos de datos y operadores básicos**
- Tema 6. Sentencias de Control



Tema 5

Tipos de datos y Operadores Básicos

Fundamentos de la Programación

Conocimientos y Habilidades

Después de este tema, el alumno debería ser capaz de:

- Conocer los tipos de datos básicos y su representación interna
- Conocer las características de las **variables** y las **constantes**
- Conocer el uso de los **operadores**
- Saber evaluar **expresiones** de distintos tipos

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Problema

Calcular la distancia en línea recta entre dos puntos de un plano

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Resolución de Problemas

1. Establecer el problema con claridad
2. Describir la información de entrada y salida
3. Hacer el problema a mano, para un caso sencillo
4. Desarrollar una solución y convertirla a un programa de ordenador
5. Comprobar la solución con varios conjuntos de datos

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

1. Enunciar el problema

- ☐ Calcular la distancia en línea recta entre dos puntos de un plano

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

2. Descripción de E/S

- ☐ El segundo paso es describir cuidadosamente:
 1. la información que se proporciona para resolver el problema
 2. Identificar los valores a calcular

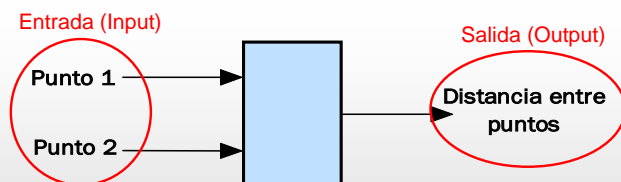


Diagrama E/S (I/O)

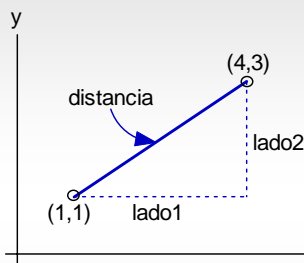
Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

3. Ejemplo a mano

- ☐ El tercer paso es hacer el problema a mano (o con calculadora) usando un pequeño conjunto de datos
- ☐ No se debe pasar al siguiente paso hasta que este no esté completo

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Distancia entre dos puntos



$p_1 = (1,1); \quad p_2 = (4,3)$

$$\begin{aligned} distancia &= \\ &= \sqrt{(lado_1)^2 + (lado_2)^2} = \\ &= \sqrt{(4-1)^2 + (3-1)^2} = \\ &= \sqrt{13} = \\ &= 3.61 \end{aligned}$$

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

4. Desarrollo del algoritmo

- Se descompone en problema en pasos:
- 1. Dar valores a los dos puntos
- 2. Calcular la distancia de los dos lados del triángulo rectángulo generado por los dos puntos
- 3. Calcular la distancia entre los dos puntos, que es igual a la hipotenusa del triángulo
- 4. Imprimir la distancia entre los dos puntos

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

- Para que la solución sirva para cualquier pareja de puntos, necesitamos algo que nos permita almacenar diferentes valores:

variable

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

¿Qué necesitamos almacenar?

- Constantes: PI, 3.14159
- Números
 - Enteros (coordenadas, ...)
 - Reales (coordenadas, distancia, ...)
- Caracteres

Tipos de datos en C:
Variables
Constantes

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

¿Qué necesitamos representar?

- ❑ Fórmulas
- ❑ Otras expresiones matemáticas

Operadores en C

Expresiones en C

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Tipos de dato

- ❑ **Constantes:** mantienen su valor durante todo el programa

- ❑ **Variables:** pueden modificar su valor en algún momento de la ejecución del programa

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Resumen de tipos de datos

Datos básicos	Numéricos	Entero		
		Real		
	Carácter			
Dato derivado	Lógico			
	Puntero			
	Internos	Estáticos	Lineales	Tabla
		Dinámicos	Lineales	Lista
				Pila
			No lineales	Cola
				Árbol
Datos estructurados	Externos	Archivo		
		Base de Datos		
	Compuesto	Estructura de datos o registro		

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática



La despensa del cocinero



La 'despensa' del ordenador

Una variable es una **posición de memoria** a la que se asigna un **identificador**, y puede almacenar un **dato**

Memory	
Address:	0x12FF78
0012FF78	00
0012FF7A	00
0012FF7C	00
0012FF7E	00
0012FF80	00
0012FF82	12
0012FF84	09
0012FF86	40
0012FF88	01
0012FF8A	00
0012FF8C	10
0012FF8E	C8
0012FF90	50

Dirección de memoria

Valor almacenado

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

¿Qué es una *variable*?

- Registro en memoria etiquetado con un nombre (**identificador**)
- A toda variable que se use en un programa, se le asocia (generalmente al principio del programa) un **tipo de dato** específico.
- Un tipo de dato define todo el posible rango de valores que una variable puede tomar al momento de ejecución del programa y a lo largo de toda la vida útil del propio programa.

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Variables

- Almacena un valor de un cierto tipo de dato
- El dato puede ser modificado (¡es variable!)
- El nombre (etiqueta) ha de ser un identificador válido (carácter alfanumérico ó _)

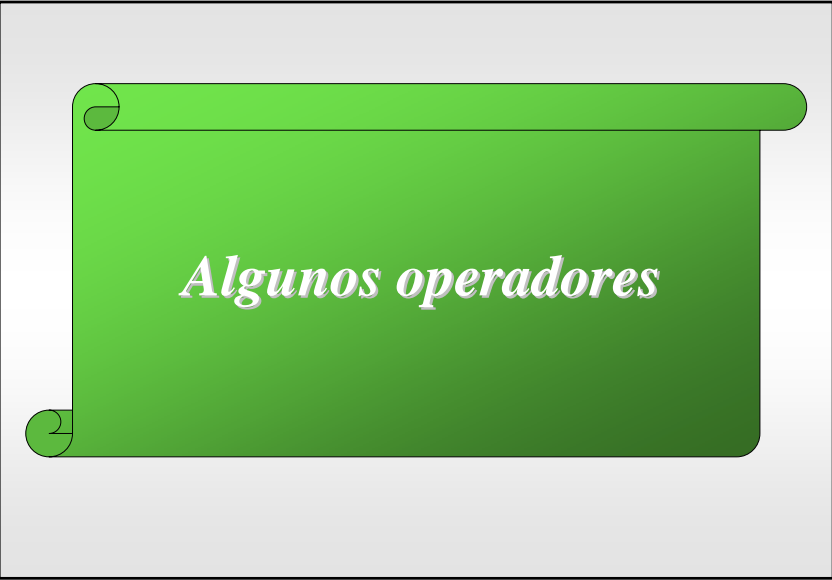
salario dia_de_la_semana 40ros
edad_alumno _fax

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Declaración de variables

- Es una sentencia que proporciona información sobre la variable:
`<tipo> <nombre_variable>;`
- Ejemplos:
`long dNumero;`
`double HorasAcumuladas;`
`float HorasPorSemana;`
- Sólo se pueden definir al principio de un archivo o bloque de código, SIEMPRE ANTES DE CUALQUIER SENTENCIA

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática



Operadores básicos

- Asignación
- Aritméticos
- Operador de tamaño
- Operador de dirección, &

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Operadores aritméticos

Monarios	Signo negativo	-
	Incremento	++
	Decremento	--
Binarios	Suma	+
	Resta	-
	Multiplicación	*
	División	/
	Módulo	%

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Operadores de asignación

- ❑ Operación: cargar una variable con el valor de una expresión
- ❑ Tipos:
 - Simple
 - Compuesto

Operador de asignación simple

- ❑ Utiliza el símbolo =
- Formato**
- Variable = expresión;

La variable situada a la **izquierda** se cargará con el **valor resultante** de la expresión situada a su derecha

Ejemplos

```
Media = 3.2;
Media = suma / num;
```

Operador de asignación compuesto

- ❑ Utiliza dos símbolos:
- =
- + - / % << >> & ^ |

Formato

Variable Símbolo_op = Expresión;

Equivale a

Variable = variable Símbolo_op Expresión;

Operador de asignación compuesto

Ejemplo

```
total /= importe + 100;
equivale a
total = total / (importe + 100);
```

Ejemplo

```
resul *= num - 10;
equivale a
resul = resul * (num - 10);
```

Operador de tamaño

- ❑ Se utiliza para: obtener la longitud en bytes de una variable o de un especificador de tipo de dato.
- ❑ Emplea la palabra reservada `sizeof`

Ejemplo

```
sizeof(float) /* tamaño tipo float */  
sizeof cant /* tamaño variable cant */
```

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Operador dirección de (&)

- ❑ Obtiene la dirección de memoria donde está ubicada una variable

Ejemplo

```
int total;  
&total; /* direccion de la variable total*/
```

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Operador coma

- ❑ Separa dos expresiones dentro de una expresión total

- ❑ Ejemplo:

```
int a, b;  
float x=0, y=4;
```

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Inicialización de variables

- ❑ En la inicialización se le asigna un **valor inicial** a la variable
- ❑ Las variables se pueden inicializar:
 - A la vez que se declaran
`<tipo> <nombre_variable> = <valor inicial>;`
 - Después de la declaración
`char barra;
....
barra = 'S';`

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Datos básicos

- ❑ Numéricos
 - Enteros
 - Reales
- ❑ Carácter
 - Simple
 - Cadena

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Numérico entero

Representación interna **con signo**

Enteros positivos



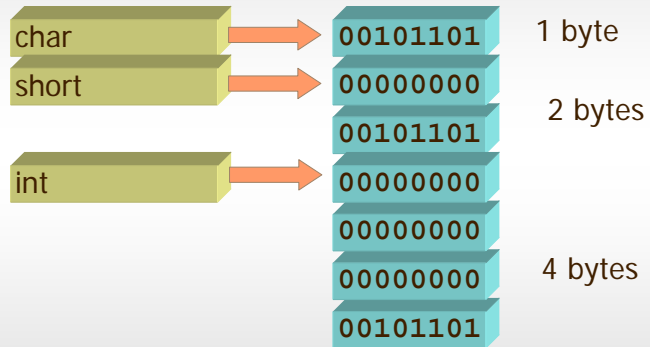
Enteros negativos (complemento a 2)



Bit de signo

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Variables enteras con signo



Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Declaración de enteros con signo

- `char` `caracter;`
- `short int` `entero_corto;`
- `int` `entero;`
- ❑ Inicializando:
 - `char` `carácter=12;`
 - `short int` `entero_corto=24;`
 - `int` `entero=-14;`

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Numérico entero

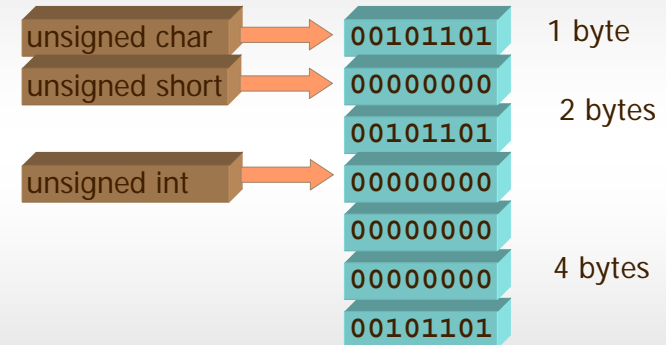
- Números enteros con signo o sin signo:
 - -47, 1025, +9

Representación interna sin signo



Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Variables enteras sin signo



Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Declaración de enteros sin signo

- `unsigned char character;`
- `unsigned short int entero_corto;`
- `unsigned int entero;`
- Inicializando:
 - `unsigned char character=12;`
 - `unsigned short int entero_corto=24;`
 - `unsigned int entero=14;`

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Numérico real

- Se emplea para representar:
 - Números con parte decimal
 - Números muy pequeños
 - Números muy grandes
- Notaciones:
 - Punto decimal: -82.75 470.05
 - Científica/exponencial: *mantisa**Ecaracterística*

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Numérico real: científica

- ❑ Número = mantisa * Base^{exponente}
- ❑ Mantisa: número real
- ❑ Característica: exponente
- ❑ Ejemplos:

2.5E3	Equivale a	2.5 x 10 ³
-0.75E-2	Equivale a	-0.75 x 10 ⁻²

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Numérico real: científica

Estándar IEEE

Base	2
Exponente	Exceso a 2 ⁿ⁻¹ -1
Mantisa	binario puro

Para simple precisión (32 bits)		Para doble precisión (64 bits)	
Signo	1 bit (posición 31)	Signo	1 bit (posición 63)
Exponente	8 bits (posiciones 23 a 30)	Exponente	11 bits (posiciones 52 a 62)
Mantisa	23 bits (posiciones 0 a 22)	Mantisa	52 bits (posiciones 0 a 51)

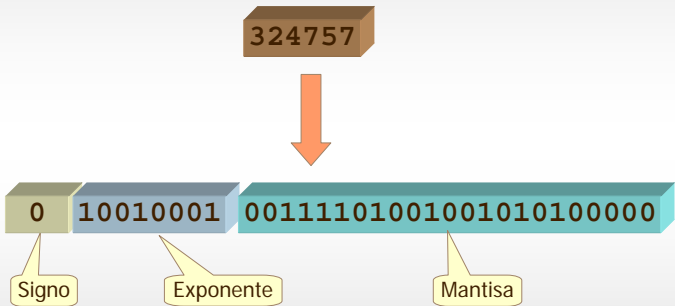
Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Ejemplo not. científica

- ❑ Número: 324757
 - ❑ Binario: 1001111010010010101
 - ❑ Formato exponencial:
1.001111010010010101 x 2¹⁸
 - ❑ Signo: 0
 - ❑ Característica:
127 + 18 = 145 (10010001)
 - ❑ Mantisa fraccionaria:
00111101001001010100000
- 010010001 00111101001001010100000
0x489E92A0

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Ejemplo not. científica



Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Variables reales

- ❑ float: simple precisión (4 bytes)
- ❑ double: doble precisión (8 bytes)
- ❑ long double: doble precisión (8 bytes)

Carácter

- ❑ Caracteres dígitos: '5', '1'
- ❑ Caracteres alfabéticos: 'H', 'p'
- ❑ Caracteres especiales: '?', '+'

también utiliza char

Carácter: código ASCII

ASCII: "American Standard Code for Information Interchange"

Carácter	Código ASCII	Representación interna
'A'	65	01000001
'9'	57	00111001
'{'	123	01111011

Resumen de tipos de datos

Tipo de dato	Byte	C	C++	Java
Entero	1	char	char	byte
Entero	2	short int	short [int]	short
Entero	4	int	int	int
Entero	4	long int	long [int]	long (8)
Real (p. simple)	4	float	float	float
Real (p. doble)	8	double	double	long
Real (p. máx.)	10	long double	long double	
Lógico	1	---	bool	boolean
Carácter	1	char	char	char (2)
Cadena caracteres	?	---	String*	String*

void: crea una variable sin tipo definido
unsigned: permite crear tipos sin signo (sólo para enteros)
*: son objetos

Rango de tipos de dato

Tipo	Datos almacenados	Bits	Con signo -2^{n-1} , $2^{n-1}-1$	Sin signo 0 , 2^n-1
char	Caracteres	8	-128 a 127	0 a 255
short	Entero corto	16	-32.768, 32.767	0, 65535
int	enteros	32	-2.147.483.648, 2.147.483.647	0, 4.294.967.295
long	enteros largos	32	-2.147.483.647, 2.147.483.647	0, 4.294.967.295
float	Nums. reales (coma flotante)	32	$-3.4 \times 10^{+38}$, $3.4 \times 10^{+38}$	
double	Nums. reales (doble precisión)	64	$-1.7 \cdot 10^{+308}$, $1.7 \cdot 10^{+308}$	

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Constantes

Constantes literales

- Constantes enteras
- Constantes reales
- Constantes carácter
- Constantes cadena

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Constantes literales

Constantes enteras

- No utilizar signos de puntuación: 123456

Tipo long:	1024L
Tipo unsigned	4352U

Formato decimal	123
Formato octal	0777
Formato hexadecimal	0xFF3A

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Constantes literales

Constantes reales

- Notación en coma flotante: 3.14159
- Notación exponencial:
 $4.5\text{E}4, -3.2\text{E}-5 \Rightarrow 45000, -0.000032$

Constantes carácter

- Es un carácter ASCII encerrado entre comillas simples

'A' 'b' 'c' '5'

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Constantes literales

Constantes cadena

- Secuencia de caracteres encerrados entre comillas dobles
- "123" "12 de octubre 1492" "esto es una cadena"
- Se representan como una serie de caracteres ASCII más un carácter nulo (\0)

Una constante de carácter se encierra entre comillas simples, y las constantes de cadena se encierran entre dobles comillas

'z' es distinto de "z"

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Constantes Definidas (Simbólicas)

- Las constantes pueden recibir nombres simbólicos

En C:

```
#define PI 3.141592  
#define AVOGADRO 6.023e23
```

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Solución final

```
/*-----  
Programa 1 tema 5  
Este programa calcula la  
distancia entre dos puntos.  
-----*/  
#include <stdio.h>  
#include <math.h>  
  
int main(void)  
{  
    /* Declaro e inicializo variables */  
    double x1=1, y1=5, x2=4, y2=7;  
    double lado1, lado2, distancia;  
  
    /* Calculo lados del triangulo */  
    lado1 = x2 - x1;  
    lado2 = y2 - y1;  
    distancia = sqrt(lado1*lado1 + lado2*lado2);  
  
    /* Muestro distancia */  
    printf("La distancia entre los dos puntos es "  
           "%5.2f \n", distancia);  
  
    /* finalizo el programa */  
    return 0;  
}
```

Función scanf()

- ❑ Permite introducir valores por teclado
- ❑ Ejemplo: `scanf("%d", &x);`
- ❑ El primer argumento es una cadena de control que especifica los tipos de las variables que se van a introducir por teclado
- ❑ Los restantes argumentos son posiciones de memoria que corresponden a los especificadores de la cadena de control

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Función scanf()

```
scanf("%lf %c", &distancia, &longitud);
```

- ❑ Lee dos valores por teclado (separados al menos por un espacio) y los convierte en un valor doble y en un valor carácter respectivamente.
- ❑ Ejercicio: ¿Cómo se debe modificar la solución obtenida para cualquier pareja de puntos?

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Deberes (por parejas)

1. Escribe un programa para convertir euros a pesetas y viceversa
2. Escribe un programa que calcule el área de la superficie de una esfera de radio r

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

