



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FCFM

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

UNIDAD TEMÁTICA II

TIPOS DE DATOS Y

OPERADORES

PROPÓSITO

Conocer los elementos básicos para la construcción de instrucciones simples en C.

Ver como combinar estos elementos para formar instrucciones mas grandes

CARACTERES ESPECIALES.

Lenguaje C utiliza como bloques de construcción:

Letras mayúsculas: A, B, C, ... , Z.

Letras minúsculas: a, b, c, ... , z.

Dígitos: 0, 1, 2, ... , 9.

Ciertos caracteres especiales: #, \$, %, &, ', ", ~, \, |, ^, &tilde, {, }, [,], <, >, ±, etc.

Lenguaje C, utiliza ciertas combinaciones de estos caracteres como:

`\b` Retroceso de espacio

`\t` Tabulador horizontal

`\n` Salto de línea

`\v` Tabulador vertical, etc.

TIPOS DE DATOS.

Existen cinco tipos de datos básicos en C y C++: carácter, entero, coma flotante, coma flotante de doble precisión y sin valor:

Los valores del tipo char se usan normalmente para guardar valores definidos en el conjunto de caracteres ASCII, así como cualquier cantidad de 8 bits.

Las variables de tipo int se usan para guardar cantidades enteras.

Las variables de tipo float o double se usan para guardar números reales (los números reales tienen un componente entero y uno fraccionario).

El tipo void tiene tres usos:

1. Para declarar explícitamente una función como que no devuelve valor alguno.
2. Para declarar explícitamente una función sin parámetros.
3. Para crear punteros genéricos.

C y C++ soportan varios tipos agregados, incluyendo las estructuras, las uniones, los campos de bits, las enumeraciones y los tipos definidos por el usuario.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FCFM

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

La siguiente tabla muestra los tipos de datos básicos, su tamaño en bytes y el rango de valores para los cuales están definidos.

TIPO	TAMAÑO EN BYTES	RANGO
char	1	0 a 255
int	2	-32768 a 32768
float	4	3.4E-38 a 3.4E+38
double	8	1.7E-308 a 1.7E+308
void	0	Sin valor



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

MODIFICADORES DE TIPO.

A excepción del tipo void, los tipos de datos básicos pueden tener distintos modificadores precediéndolos.

Un modificador se usa para alterar el significado del tipo base para que se ajuste mas precisamente a las necesidades de cada momento. A continuación se muestra la lista de los modificadores:

- ✓ **signed**
- ✓ **unsigned**
- ✓ **long**
- ✓ **short**

Los modificadores signed, unsigned, short y long se pueden aplicar a los tipos base entero y carácter.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Sin embargo, long también se puede aplicar a double. El uso del signed es redundante pero es admisible.

La siguiente tabla muestra todas las combinaciones que se ajustan al estándar ANSI C junto con sus rangos y longitudes suponiendo palabras de 16 bits (2 bytes).

TIPO	TAMAÑO	RANGO
char	1	-128 a 127
unsigned char	1	0 a 255
signed char	1	-128 a 127
int	2	-32768 a 32767
unsigned int	2	0 a 65535



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

signed int	2	-32768 a 32767
short int	2	-32768 a 32767
unsigned short int	2	0 a 65535
signed short int	2	-32768 a 32767
long int	4	-2147483648 a 2147483647
unsigned long int	4	0 a 4294967295
signed long int	4	-2147483648 a 2147483647
float	4	3.4E-38 a 3.4E+38
double	8	1.7E-308 a 1.7E+308
long double	10	1.7E-512 a 1.7E+512



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

La diferencia entre enteros con signo y sin signo está en como se interpreta el bit más significativo del entero.

Si se especifica un entero con signo, el compilador genera código que asume que el bit más significativo va a ser usado como un indicador de signo.

Si el indicador del signo es 0, entonces el número es positivo; si es 1, el número es negativo.

Por ejemplo:

127 en binario es 0000 0000 0111 1111₂

-127 en binario es 1000 0000 0111 1111₂



bit del signo



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FCFM

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

IDENTIFICADORES.

Los nombres usados para referirse a las variables, las funciones, las etiquetas, las constantes y otros objetos definidos por el usuario se conocen como identificadores.

La longitud de un identificador puede variar entre 1 y 31 caracteres para C y no hay limite para C++.

Las reglas para formar identificadores son:

- El primer carácter debe ser letra o un símbolo de subrayado guion bajo.
- Los demás caracteres pueden ser letras, números o símbolo de subrayado.

- Los identificadores no pueden ser iguales a las palabras claves de C.
- Los identificadores no deben tener el mismo nombre que alguna función ya escrita o que se encuentren en las bibliotecas de C.

Por ejemplo:

Correcto	Incorrecto
Cont	1cont
prueba23	hola!
balance_total	balance...total
_total	x y
prom	num-2

En C las minúsculas y las mayúsculas se tratan como distintas, así: cuenta, Cuenta, CUENTA representan a tres identificadores distintos.

CONSTANTES EN C

Las constantes en C se refieren a valores fijos que no pueden ser alterados por el programa. Pueden ser del cualquier tipo de dato.

En C, existen cuatro tipos de constantes: literales, definidas, declaradas y enumeradas.

Constantes literales .- o constantes, en general, se clasifican en cuatro grupos: enteras, caracteres, de coma flotante, y de cadena.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Constantes enteras.- es un número con un valor entero, consiste en una secuencia de dígitos.

Se pueden representar en tres sistemas numéricos diferentes:

- Formato decimal: 123, -123, 0, 458
- Formato octal: 0123, 0777, 015 (**están precedidos de la cifra 0**)
- Formato hexadecimal: 0x125, 0x2A, 0xABC (**comienza con 0x**)

Se pueden utilizar los sufijos L (l) que significa long o bien U (u) que significa unsigned, por ejemplo:

3452L

1024L

123U

123UL

Constantes reales.- una constante de coma (punto flotante) representa un número real, que contiene punto decimal o un exponente o ambos. Por ejemplo:

82.125, 0.123, 83., 47e-4, 1.25E12, 61.e+4

Existen tres tipos de constantes: float, double, y long double.

Nota

- ✓ Las constantes enteras y reales representan números, por lo tanto se les denomina constantes numéricas.
- ✓ No pueden incluir comas ni espacios en blanco.
- ✓ Pueden ir precedidas del signo (-).

Constantes de carácter.- una constante de carácter (char) es un carácter del código ASCII, encerrado entre comillas simples. Por ejemplo:

‘A’, ‘a’, ‘+’, ‘?’, ‘9’, ‘ ‘, ‘#’, etc.

A cada carácter le corresponde su código numérico del carácter. Existen caracteres especiales, denominados de escape, que suelen representar caracteres tales como la comilla, doble comilla, signo ?, retorno de carro, etc.

Constantes de cadena.- es una secuencia de caracteres o ninguno encerrados entre dobles comillas. Por ejemplo:

“HOLA”, “esto es una cadena”, “ ”, “U.A.N.L.”

Se puede escribir una cadena en varias líneas, terminando cada línea con “\”. Por ejemplo:

“Esto es una cadena \
que tiene dos líneas”

Constantes definidas (simbólicas).- las constantes pueden recibir nombres simbólicos mediante la directiva #define. Por ejemplo:

```
#define NUEVA LINEA \n
```

```
#define PI 3.141592
```

```
#define VALOR 100
```

C sustituye los valores `\n`, 3.141592 y 100 cuando se encuentran las constantes simbólicas NUEVA LINEA, PI, y VALOR.

Las líneas anteriores no son sentencias y, por ello, no terminan en punto y coma.

Constantes enumeradas.- permite crear listas de elementos afines. Por ejemplo:

```
enum color {rojo, naranja, amarillo, verde, azul, violeta};
```

Cuando se procesa esta sentencia, el compilador asigna un valor, a cada elemento enumerado que comienza en cero; así, rojo equivale a 0, naranja a 1, amarillo a 2, verde a 3, azul a 4, y violeta a 5.

El compilador enumera los identificadores.

Después de declarar un tipo de dato enumerado, se pueden crear variables de ese tipo, tal como con cualquier otro tipo de dato. Por ejemplo, se puede definir una variable de tipo color:

```
color colorfavorito = verde;
```



UANL
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FCFM
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Constantes declaradas.- el cualificador const permite dar nombres simbólicos a constantes. Su valor no puede ser modificado por el programa, su formato general para crear una constante es:

`const tipo nombre = valor;`

La palabra reservada volatile actúa como const, pero su valor puede ser modificado no sólo por el propio programa, sino también por el hardware o por el software del sistema.

VARIABLES

En C una variable es una posición con nombre (identificador) de memoria donde se almacena un valor de un tipo de dato. Su valor puede cambiar a lo largo de la ejecución del programa.

Declaración.- Una declaración de una variable es una sentencia que proporciona información de la variable al compilador de C. es preciso declarar las variables antes de utilizarlas. Su sintaxis es:

tipo nombre; { **Tipo.- es el nombre de un tipo de dato en C.**
Nombre.- identificador en C.

Por ejemplo: int x, char nom, float X_1 , int prom, etc.

Inicialización de variables.- Las variables se pueden inicializar a la vez que se declaran o bien después de la declaración, es decir, cada nombre de variable puede ir seguido por una expresión opcional de preparación que asigne un valor inicial. Por ejemplo:

```
char respuesta = 'S';  
int x = 2510;  
int contador = 1;  
float peso = 156.45;  
int año = 2005;  
char barra = '\\';
```



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

OPERADORES

Lenguaje C tiene un poderoso conjunto de operadores. Usándolos bien, aumenta la expresividad y eficiencia del lenguaje. Si se usan en forma equivocada, conducen a expresiones difíciles de leer o que dan resultados erróneos.

Son cuatro los tipos generales de operadores, de acuerdo con el número de operandos requeridos:

- Unarios.- requieren de un operando.
- Binarios.- requieren de dos operandos (aritméticos, relacionales, de igualdad, lógicos, de asignación de bits, y de serie).
- Terciario.- solo hay un operador el condicional (?), que utiliza tres operandos.
- De membresía.- se clasifican como expresiones postfijas



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FCFM

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Operadores aritméticos.- a través de ellos se pueden realizar operaciones aritméticas básicas.

Operador	Tipos enteros	Tipos reales
+	Suma	Suma
-	Resta	Resta
*	Producto	Producto
/	División entera: Cociente	División en punto flotante
%	División entera: residuo	

Operadores de incremento.- los operadores ++, --, denominados de incremento y decremento, suman o restan uno a su argumento respectivamente



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Incremento	Decremento
$++n, n++$	$--n, n--$
$n += 1$	$n -= 1$
$n = n + 1$	$n = n - 1$

Las sentencias $++n$ y $n++$, $--n$ y $n--$, tienen el mismo efecto.
Tiene la propiedad de que pueden utilizarse como sufijo o prefijo.

Por ejemplo:

```
int a = 1, b;
```

$b = a++;$ { Se realiza primero la asignación y después se incrementa a en una unidad.
b vale 1 y a vale 2.

$b = ++a;$ { Se incrementa a en una unidad y después se realiza la asignación.
b vale 2 y a vale 2.



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FCFM

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Operadores de asignación.- una asignación como $n = 10$, es una expresión y produce un valor como cualquier otra expresión; el valor devuelto es el mismo que el valor asignado de la variable.

Formato: identificador = expresión

Donde expresión puede ser: una constante, una variable, una expresión, una expresión compuesta.

Además del operador de asignación, $=$, existen otros operadores de asignación relativa. Así el operador $+=$:

Por ejemplo:

cuenta += 10; equivale a cuenta = cuenta + 10;



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FCFM

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Operador	Sentencia abreviada	Sentencia no abreviada
+=	$m += n$	$m = m + n$
-=	$m -= n$	$m = m - n$
*=	$m *= n$	$m = m * n$
/=	$m /= n$	$m = m / n$
%=	$m \% = n$	$m = m \% n$

Operadores relacionales.- Lenguaje C no tiene tipos de datos lógicos o booleanos como el lenguaje Pascal, para representar los valores de verdadero y falso.

En su lugar se utiliza el tipo int para este propósito, con el valor entero 0 que representa a “falso” y distinto de cero a “verdadero”.

Operadores tales como \geq , $<$, $=$ que comprueban una relación entre dos operandos se llaman operadores relacionales y se utilizan en expresiones de la forma:

expresión₁ operador_relacional expresión₂

Los operadores relacionales se usan normalmente en sentencias de selección (if), o de iteración (while, for), que sirve para comprobar una condición.

Utilizando operadores relacionales se realizan operaciones de igualdad, desigualdad y diferencias relativas.

Operador	Significado	Ejemplo
$=$	Igual a	$a = b$
\neq	No igual	$a \neq b$
$>$	Mayor que	$a > b$
$<$	Menor que	$a < b$
\geq	Mayor o igual que	$a \geq b$
\leq	Menor o igual que	$a \leq b$

Los operadores relacionales tiene menor prioridad que los operadores aritméticos.

Operadores lógicos.- Estos operadores se utilizan con expresiones para devolver un valor verdadero (distinto de cero), o un valor falso (cero).

Los operadores lógicos se denomina también operadores booleanos.

Los operadores lógicos en C son:

- ✓ not (!) .- produce falso si su operando es verdadero o viceversa.
- ✓ and (&&) .- produce verdadero solo si ambos operandos son verdaderos; si cualquiera de los operandos es falso produce falso.
- ✓ or (||) .- produce verdadero si cualquiera de los operandos es verdadero y produce falso si ambos son falsos.

Operador	Operación lógica	Ejemplo
Negación !	No lógica	$!(x \geq y)$
Y lógica &&	operando_1 && operando_2	$m < n \ \&\& \ i > j$
O lógica 	operando_1 operando_2	$m == 5 \ \ n != 10$

Nota.

El operador ! Tiene prioridad mas alta que &&, que a su vez tiene mayor prioridad que ||

Operadores de manipulación de bits

Los operadores de manipulación o tratamiento de bits ejecutan operaciones lógicas sobre cada uno de los bits de los operandos.

Estas operaciones son comparables en eficiencia y en velocidad a sus equivalentes en Lenguaje Ensamblador.

Los operadores de manipulación de bits se aplican solo a variables y constantes char, int y long, no a datos float.

La siguiente tabla muestra los operadores de manipulación de bits:



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Operador	Operación
&	Y (AND) lógica de bit a bit
	O (OR) lógica de bit a bit
^	OR exclusivo de bit a bit (XOR)
~	Complementos a uno
<<	Desplazamientos de bits a la izq.
>>	Desplazamientos de bits a la der.

Nota : los operadores de desplazamiento >>, <<; efectúan un desplazamiento a la derecha (>>) o a la izquierda (<<) de n posiciones de los bits del operando, siendo n un número entero

Operadores de dirección

Son operadores que permiten manipular las direcciones de los objetos:

Operador	Acción
*	Lee o modifica el valor apuntado por la expresión. Se corresponde con un apuntador y el resultado es del tipo apuntador
&	Devuelve un apuntador al objeto utilizado como operando que debe ser una variable dotada de una dirección de memoria. El resultado es un apuntador de tipo idéntico al del operando
•	Permite acceder a un miembro de un objeto agregado (unión, estructura o clase)
->	Accede a un miembro de un objeto agregado (unión, estructura o clase) apuntado por el operando de la izquierda.

Operador condicional ?

Es un operador ternario que devuelve un resultado cuyo valor depende de la condición comprobada. Tiene asociatividad a derecha (derecha a izquierda).

Al ser un operador ternario requiere de tres operandos. Se utiliza para reemplazar a la sentencia if – else lógica en algunas situaciones.

Formato:

`expresión_c ? Expresión_v : expresión_f;`

Operador serie (operador coma)

El operador coma (,) permite combinar dos o mas expresiones separadas por comas en una sola línea. Se evalúa primero la expresión de la izquierda y luego las restantes expresiones de izquierda a derecha.

Por ejemplo:

`i++, j++;`

equivale a

`i++;
j++;`

`i++, j++, k++`

equivale a

`i++;
j++;
k++;`

Operadores especiales

Operador () .- es el operador de llamada a funciones.

Sirve para encerrar los argumentos de una función, efectuar conversiones explícitas de tipo, indicar en el seno de una declaración que un identificador corresponde a una función y resolver los conflictos de prioridad entre operadores.

Operados [] .- sirve para dimensionar los arreglos y designar un elemento de un arreglo.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FCFM

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Operador sizeof

Con frecuencia su programa necesita conocer el tamaño en bytes de un tipo de dato o variable.

C proporciona el operador sizeof, que toma un argumento, bien un tipo de dato o bien el nombre de una variable (escalar, arreglo, registro, etc.).

Sintaxis:

sizeof(nombre_variable)
sizeof(tipo_dato)
sizeof(expresión)



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FCFM

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Evaluación de expresiones

Una expresión, es una secuencia de operadores y operandos que especifican un cálculo.

Tipos de expresiones:

1. Simples.- son aquellas que involucran operadores con igual prioridad y de operandos con el mismo tipo.
2. Mixtas.- son aquellas que involucran operadores de diferente prioridad y de operandos con tipos diferentes.

Para evaluar las expresiones es necesario tomar en cuenta la precedencia y la asociatividad de los operadores involucrados.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICA



FCFM

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Prioridad (precedencia) y asociatividad

La prioridad o precedencia de operadores determina el orden en que se aplican los operadores a un valor.

Para evaluar expresiones que involucran diferentes operadores, se toma en cuenta lo siguiente:

- ✓ Si dos operadores se aplican al mismo operando, el operador con mayor prioridad se aplica primero.
- ✓ Todos los operadores del mismo grupo tienen igual prioridad y asociatividad.
- ✓ La asociatividad izq – der significa aplicar el operador más a la izquierda primero, y en la asociatividad der – izq se aplica primero el operador más a la derecha.