



UNIVERSIDAD NACIONAL DE INGENIERIA



CENTRO DE TECNOLOGIAS DE INFORMACION Y COMUNICACIONES

Programación en C++

Sesión 2: Introducción

Unidad



Sesión 2: Elementos de un programa en C++

Contenidos

- a) Tipos de Datos Estándar
- b) Constantes
- c) Variables
- d) Sentencias
- e) Entrada y Salida Básica

Aprendizajes esperados

- Aprende a crear un programa en C++

TIPOS DE DATOS ESTÁNDAR

TIPOS FUNDAMENTALES

➤ TIPO ENTERO

■ *CON SIGNO*

Tipo	Rango	Tamaño (bytes)
int	-2147483648...2147483647	4
long	-2147483648...2147483647	4
long int	-2147483648...2147483647	4
char	-128...127	1
enum	-32768...32767	2
short	-32768...32767	2
short int	-32768...32767	2

- *SIN SIGNO:*

Tipo	Rango	Tamaño (bytes)
unsigned short	0...65535	2
unsigned int	0...4294967295	4
unsigned long	0...4294967295	4
unsigned char	0...255	1

➤ **TIPO REAL** (Punto Flotante)

Tipo	Rango	Tamaño (bytes)
float	$\pm 3.4e-38 \dots \pm 3.4e+38$	4
double	$\pm 1.7e-308 \dots \pm 1.7e+308$	8
long double	$\pm 3.4e-4932 \dots \pm 1.1e+4932$	12

TIPOS DERIVADOS

Son contruidos a partir de los fundamentales

- Punteros
- Estructuras
- Uniones
- Arreglos
- Funciones
- Clases

CONSTANTES

Una constante es un valor que no cambia durante la ejecución de un programa y es fijado por el compilador.

❑ CONSTANTES ENTERAS

EJEMPLO:

- 123 (formato decimal)
- 0264 (formato octal)
- 0xBE3F (formato hexadecimal)
- 1132U (formato entero sin signo)
- 1431L (formato entero largo con signo)
- 325UL (formato entero largo sin signo)

➤ CONSTANTES REALES

EJEMPLOS:

- 14.53
- 77.313423
- 0.521
- .431
- 1.23e5 (notación científica del numero 1.23×10^6)
- -12.3e-3 (notación científica del numero -12.3×10^{-3})

➤ CONSTANTES DE UN SOLO CARÁCTER

EJEMPLOS:

- 'M'
- 'm'
- 'E'
- '\n'
- '\r'

➤ CONSTANTE DE CARACTERES

Están formadas por una secuencia de caracteres encerrados entre dobles comillas.

EJEMPLOS:

- “mi cadena”
- “orson welles”

➤ CONSTANTE DECLARADAS

❑ TIPO const

Formato

const *tipo nombre* = valor

EJEMPLOS:

- const int **semana** = 7;
- const double **pi** = 3.1415;

VARIABLES

Una Variable es un valor que puede cambiar a durante la ejecución de un Programa. Las variables deben ser declaradas antes de ser usadas.

DECLARACION DE VARIABLES

(a) *Formato para una sola variable*

tipo identificador

donde:

- *tipo* : es el tipo de dato de la variable
- *identificador* : nombre de la variable

EJEMPLOS:

```
int a;
```

```
double area;
```

(b) *Formato para mas de una sola variable*

tipo identif1, identif2,...

donde:

- *tipo* : es el tipo de dato de las variables
- *identif1, identif2, ...* : nombres de las variables

EJEMPLOS:

int a, b, c;

double area, longitud;

SENTENCIAS

Una sentencia es la unidad ejecutable más pequeña del programa. Están compuestas por palabras reservadas, expresiones, declaraciones o llamadas a funciones.

➤ SENTENCIA SIMPLE

Es aquella que termina con un punto y coma (;).

```
int m ;  
double area, longitud ;
```

❑ SENTENCIA DE ASIGNACIÓN

Permite almacenar valores a las variables. Se utiliza el operador =.

```
m = 18;  
area = 98.53;
```

❑ INICIALIZACIÓN DE VARIABLES

Consiste en asignarle un valor a la variable en su declaración.

```
int k=7;
```

```
double fmin=30, fmax=20000;
```

➤ SENTENCIA COMPUESTA

Es una colección de sentencias simples encerradas entre llaves

{ : llave de inicio

} : llave de fin

EJEMPLO:

```
{
```

```
int m, n=3, p;
```

```
p=2*n;
```

```
m=n+p;
```

```
}
```

*A las sentencias
compuestas se les conoce
también con el nombre de
bloques.*

ENTRADA Y SALIDA EN C++

C++ utiliza clases para la entrada y salida de datos a través de flujos (streams), los cuales son secuencias de bytes con una colección de rutinas de inserción (<<) y extracción (>>) bytes.

SALIDA:

Flujo : cout (console out)

Operador : << (operador de inserción)

Formato :

```
cout<<elemento1<<elemento2<<...<<elementoN;
```

donde:

elemento1,..., elementoN : puede ser una variable,
constante o llamada a
función.

EJEMPLO 1:

```
int num1 = 4, num2;  
num2 = num1 + 7;  
cout<<"El resultado es = "<<num2;
```

Salida:

El resultado es = 11

cursor

espacio en blanco

EJEMPLO 2:

```
double f = 321.231;  
cout<<"\nLa frecuencia es = "<<f<<"hertz";
```

Salida:

La frecuencia es = 321.231hertz

Línea en blanco (debido a la presencia de la sec. de escape '\n') al inicio de la cadena "La frecuencia es = "

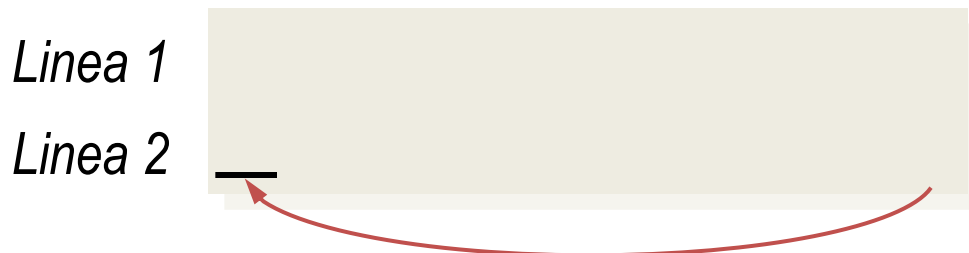
NOTA

La secuencia de escape `'\n'` realiza dos operaciones:

➤ cambio de línea



➤ retorno de carro.



EJEMPLO: La sentencia

```
cout<<"Universidad\n"<<"Nacional"<<"\nde Ingeniería\n";
```

Producirá la salida:

```
Universidad
Nacional
de Ingeniería
```

—

ENTRADA

Flujo : cin (console out)

Operador : >> (operador de extracción)

Formato :

```
cin>>variable1>>variable2>>...>>variableN;
```

donde:

variable1 , ... , *variableN* : debe ser una variable.

EJEMPLO 1:

```
int num1, num2;  
cout<<"Ingrese num1=";  
cin>>num1;  
cout<<"Ingrese num2=";  
cin>>num2;
```

EJEMPLO 2:

```
int f;  
cout<<"Ingrese frecuencia=";  
cin>>f;  
cout<<"\nLa frecuencia es = "<<f<<"hertz";
```