

# TEMA: INTRODUCCION AL C++ (SESION 03)

---

SEMESTRE 2024-1

## LOGRO DEL APRENDIZAJE:

Implementar programas secuenciales haciendo uso correcto de los tipos de datos, las funciones predefinidas y las instrucciones de entrada y salida de datos.

### KEYWORDS:

Tipos de datos, operadores , expresiones, variable, constante, programa, función

- Introducción al C++ (Sesión 1)
  - ✓ Tipos de datos
  - ✓ Definición de variables y constantes
  - ✓ Operadores Aritméticos y Asignación
  - ✓ Instrucciones de entrada y salida
  - ✓ Estructura de un programa en C++
- Función en C++ (Sesión 2)
  - ✓ Definición de una función
  - ✓ Funciones propias del C++
  - ✓ Solución de un problema: sin función y con función
  - ✓ Sintaxis de una función
  - ✓ 4 Formas de realizar una función
  - ✓ Función con parámetro por valor o copia

## 1.2 INTRODUCCION AL C++

### TIPOS DE DATOS

Un tipo de datos es la propiedad de un valor que determina su dominio (qué valores puede tomar)



Por ejemplo, si queremos saber de una persona:

El nombre es un conjunto de **caracteres**.

La edad es un número **entero**.

La estatura es un número **decimal**.

El estado civil es un **carácter**.

Como podemos apreciar tenemos diferentes tipos de datos, números enteros y números decimales, caracteres.

## 1.2 INTRODUCCION AL C++



Tipo	Descripción	Rango	Ejemplo
<u>bool</u>	Valor binario verdadero o falso.	true False	bool dato = false; dato = true;
char	Valor entero que representa un carácter de la tabla ASCII	-128 a 127 ó 0 a 255 compilado con /J	char letra = 'A'; letra = '\n'; letra = 65;
short	Valor entero de 2 bytes	-32,768 a 32,767	short x = 94; x = -54;
int	Valor entero de 4 bytes	-2,147,483,648 a 2,147,483,647	int x = 1598; x = -988574;
unsigned int	Valor entero positivo de 4 bytes	0 a 4,294,967,295	unsigned int x = 9887; x = 98745;
long long	Valor entero de 8 bytes	-9,223,372,036,854,775,808 a 9,223,372,036,854,775,807	long long x = 684574; x = -998564;
float	Valor decimal de 4 bytes	3.4E +/- 38 (7 dígitos)	float x = 45.6; x = -98.58;
<u>double</u>	Valor decimal de 8 bytes	1.7E +/- 308 (15 dígitos)	double x = 9878.568; x = -98745.668;
void	Tipo de dato nulo. Representa la ausencia de valor.		

## 1.2 INTRODUCCION AL C++

### DEFINICIÓN DE IDENTIFICADORES: VARIABLES Y CONSTANTES

Las variables son espacios en memoria que permiten almacenar y modificar un valor en cualquier punto del programa.

Las constantes son espacios en memoria que permiten almacenar un valor pero no modificarlo.



Las variables y constantes tienen que especificar a que tipo de dato pertenecen para que el compilador sepa cómo interpretarlos.

Las variables y constantes deben poseer un nombre único dentro del programa y es obligatorio declararlas antes de poder utilizarlas.

## 1.2 INTRODUCCION AL C++

### REGLAS A SEGUIR PARA SU DEFINICIÓN

El primer carácter debe ser una letra o el signo de subrayado "\_". Los caracteres restantes pueden ser letras, el signo del subrayado o dígitos.

No se pueden utilizar caracteres especiales en el nombre como por ejemplo (ñ, ?, , , %, etc.)

Debe ser distinto de las palabras reservadas por el lenguaje, por ejemplo, int, float, if, else, etc.

El Lenguaje C, hace diferencia entre mayúsculas y minúsculas por lo que una variable llamada Sueldo es distinta de una variable llamada sueldo.



## 1.2 INTRODUCCION AL C++

### DEFINICIÓN DE UN VARIABLE EN C++

Existen tres sitios donde se pueden definir variables:

- Fuera de todas las funciones (variables globales).
- Dentro de las funciones (variables locales)
- En la definición de parámetros de funciones (parámetros formales).



Variable Global (NO es buena práctica)	Variable Local	Parámetro
<pre>#include &lt;stdio.h&gt; int x; void main() { }</pre>	<pre>#include &lt;stdio.h&gt; void main() {     int x; }</pre>	<pre>#include &lt;stdio.h&gt; void main(int x) { }</pre>



# 1.2 INTRODUCCION AL C++

## TIPOS DE UN VARIABLE EN C++

- Estáticas, tienen un tamaño asignado desde el momento en que se crea el programa.
- Dinámicas, son aquellas para las cuales se reserva espacio en el momento de ejecución del programa, sólo en la cantidad necesaria(SE DESARROLLARA DESPUES DEL PARCIAL).



Estática	Dinámica
<Tipo de dato> <Nombre variable>	<Tipo de dato> * <Nombre variable>;
int edad; bool soltero; char talla; ('S' 'M' 'L') float estatura;	#include <stdio.h> void main() { int * arreglo=new int[4]; }

## 1.2 INTRODUCCION AL C++

### DEFINICIÓN DE UNA CONSTANTE EN C++

Existen dos formas como se podría manejar:

- Sintaxis de constante :  
`const <Tipo de dato> <Nombre constante> = <valor>;`
- Sintaxis de macro :  
`#define <Nombre MACRO> <valor>`



CONSTANTE	MACRO
<code>const int ValorMaximo = 50;</code> <code>const double IGV = 18.00;</code> <code>const char PrimeraLetra = 'A';</code>	<code>#define IGV 18.00</code> <code>#define PrimerMes 3</code> <code>#define PalabraClave "Universidad"</code>

## 1.2 INTRODUCCION AL C++

### OPERADORES ARITMETICOS

Los operadores con igual nivel de prioridad se evalúan de izquierda a derecha.



Símbolo	Operación
1) ( )	Agrupación
2) * / %	Multiplicación, División, Módulo (Solo de números enteros)
3) + -	Suma, Resta

### OPERADORES DE ASIGNACION

Sirve para recuperar o guardar los valores obtenidos al realizarse o ejecutarse una expresión.

Símbolo	Operación
1) =	Asignación
2) ==	Comparación

# 1.2 INTRODUCCION AL C++

## INSTRUCCIONES DE ENTRADA Y SALIDA

- ✓ Para recibir datos desde el teclado utilizaremos el objeto **cin**.
- ✓ Para mostrar datos en la pantalla utilizaremos el objeto **cout**.

Ambos están en la librería `#include <iostream>`



SALIDA	ENTRADA
<p>El objeto <b>cout</b> puede ser utilizado de 2 formas:</p> <pre>cout &lt;&lt; "Texto a mostrar";</pre> <pre>cout &lt;&lt; "Texto" &lt;&lt; valores;</pre> <p><b>Ejemplos:</b></p> <pre>cout &lt;&lt; "UPC..., protagonistas del cambio!!!";</pre> <pre>cout &lt;&lt; "El sueldo es " &lt;&lt; sueldo;</pre> <pre>cout &lt;&lt; "La suma de A + B es: " &lt;&lt; 10 + 5;</pre>	<p>El objeto <b>cin</b> puede ser utilizado de la siguiente forma:</p> <pre>cin &gt;&gt; variable [ &gt;&gt; variable];</pre> <p><b>Ejemplos:</b></p> <pre>int num, num2;</pre> <pre>float sueldo;</pre> <pre>char letra;</pre> <pre>cin &gt;&gt; num;</pre> <pre>cin &gt;&gt; letra;</pre> <pre>cin &gt;&gt; sueldo;</pre> <pre>cin &gt;&gt; num &gt;&gt; num2;</pre>

## 1.2 INTRODUCCION AL C++

### ESTRUCTURA DE UN PROGRAMA EN C++

```
#include <iostream>
#include <conio.h>

using namespace System;
using namespace std;

void main()
{ // Entrada
  int n1,n2,suma;
  cout << "Ingresa 2 enteros :";
  cin >> n1>>n2;
  //Proceso
  suma=n1+n2;
  //Salida
  cout << "La suma es: " << suma;
  _getch();
}
```

{ Directivas del preprocesador

{ Nombres de espacio



## 1.2 INTRODUCCION AL C++

### EJEMPLO DE UN PROGRAMA EN C++

Problema : Calcular el área de un rectángulo

Datos de entrada: Largo y Ancho

Dato de salida: Área



EJEMPLO 1	EJEMPLO 2
<pre>void main() { float L, A;   cout &lt;&lt; "Ingrese el largo: ";   cin &gt;&gt; L;   cout &lt;&lt; "Ingrese el ancho: ";   cin &gt;&gt; A;   cout &lt;&lt; "El área es: " &lt;&lt; L * A;   _getch(); }</pre>	<pre>void main() { float L, A;   cout &lt;&lt; "Ingrese Largo y Ancho: ";   cin &gt;&gt; L &gt;&gt; A;   cout &lt;&lt; "El área es: " &lt;&lt; L * A;   _getch(); }</pre>
<pre>Ingrese el largo: 10 Ingrese el ancho: 5 El área es: 50</pre>	<pre>Ingrese el Largo y Ancho: 10 5 El área es: 50</pre>

## 1.3 FUNCION EN C++

### DEFINICIÓN DE FUNCION

C++ es un lenguaje modular y se puede dividir en varios módulos, cada uno de los cuales realiza una tarea determinada.

Cada módulo es un subprograma llamado función.

Las funciones sirven para:

- ✓ Realizar tareas concretas
- ✓ Simplificar el programa
- ✓ Evitar escribir el mismo código varias veces
- ✓ Dar claridad al programa



## 1.3 FUNCION EN C++

### SOLUCION DE UN PROBLEMA: SIN FUNCION Y CON FUNCION

Escribir un programa que calcule la suma de los dígitos de 2 números enteros positivos de 3 dígitos.



PROGRAMA SIN FUNCIONES	PROGRAMA CON FUNCIONES
<pre>#include &lt;iostream&gt; #include &lt;conio.h&gt; using namespace std; void main() { int c1, d1, u1, c2, d2, u2;   int n1, n2, s1, s2;   cout&lt;&lt;"Ingrese el numero1 y el numero2:";   cin&gt;&gt; n1 &gt;&gt; n2;   c1 = n1 / 100;   d1 = (n1 / 10) % 10;   u1 = n1 % 10;   s1 = c1 + d1 + u1;   c2 = n2 / 100;   d2 = (n2 / 10) % 10;   u2 = n2 % 10;   s2 = c2 + d2 + u2;   cout&lt;&lt;"La suma de dígitos es: "&lt;&lt; s1 + s2;   _getch(); }</pre>	<pre>#include &lt;iostream&gt; #include &lt;conio.h&gt; using namespace std; int SumaDigitos(int num) { int s=0, c, d, u;   c = num / 100;   d = (num / 10) % 10;   u = num % 10;   s = c + d + u;   return s; } int main() { int n1, n2, suma;   cout&lt;&lt;"Ingrese el numero1 y el numero2:";   cin&gt;&gt; n1 &gt;&gt; n2;   suma = SumaDigitos(n1) + SumaDigitos(n2);   cout&lt;&lt;"La suma de dígitos es: "&lt;&lt; suma;   _getch(); }</pre>



## 1.3 FUNCION EN C++

### FUNCION PROPIAS DEL C++



FUNCIÓN	EJEMPLO DE USO	DESCRIPCIÓN	LIBRERÍA
<code>_getch</code>	<code>char tecla = _getch();</code>	Espera a que el usuario presione una tecla y devuelve la tecla presionada.	<code>conio.h</code>
<code>toupper</code>	<code>char letra=toupper('a');</code>	Devuelve la mayúscula de un carácter	<code>ctype.h</code>
<code>pow</code>	<code>float n1 = pow(4, 3);</code>	Devuelve la potencia de 4 elevado a la 3	<code>math.h</code>
<code>sqrt</code>	<code>double raiz = sqrt(4.0);</code>	Devuelve la raíz cuadrada del número.	<code>math.h</code>

# 1.3 FUNCION EN C++

## SINTAXIS DE UNA FUNCIÓN EN C++

```
<TIPO DE DATO RETORNO> <NOMBRE FUNCIÓN> ( <PARÁMETROS> )  
{ [INSTRUCCIONES;]  
  RETURN VALOR_DE_RETORNO;  
}
```



<tipo de dato retorno>	int, float, long long, char, bool, etc. void
<nombre función>	El nombre que se le desee dar a la función siempre y cuando respete las reglas establecidas para nombrar identificadores.
<parámetros>	0, 1 o más parámetros. Cada parámetro debe especificar el tipo de dato y el nombre. Los parámetros se deben separar con comas.
return	Devuelve el valor esperado. Si la función tiene como tipo de dato retorno un int entonces el valor_De_Returno deberá ser un int. En el caso de que el tipo de dato retorno sea void bastará con poner return;

# 1.3 FUNCION EN C++



Las funciones van  
antes de la función  
main

La función **main** es  
la que se ejecuta al  
inicio del  
programa.

```
#include <iostream>
#include <conio.h>
using namespace std;

int elevaAlCuadrado(int NumeroAElevar)
{
    return NumeroAElevar * NumeroAElevar;
}

void imprimeLinea()
{
    cout<<"-----\n";
}

void main()
{
    int N, resultado;
    cout<<"Ingrese el valor de N:";
    cin>> N;
    resultado = elevaAlCuadrado (N);
    cout<<"El cuadrado de N es "<< resultado;

    imprimeLinea();
    imprimeLinea();

    resultado = elevaAlCuadrado(resultado);
    cout<<"El cuadrado de N2 es "<< resultado;
    _getch(); // espero un caracter, ejemplo:enter
}
```

Se utilizan  
las funciones  
creadas

# 1.3 FUNCION EN C++

## FORMAS DE REALIZAR UNA FUNCIÓN EN C++



Sin parámetros

Con parámetros

Sin  
retorno

```
void ImprimeSuma()
{ int a,b,suma;
  cout <<"Ingrese 2 enteros:";
  cin >>a>>b;
  suma = a + b;
  cout <<"La suma es :"<<suma;
}
```

```
void ImprimeSuma(int a, int b)
{
  int suma = a + b;
  cout <<"La suma es :"<<suma;
}
```

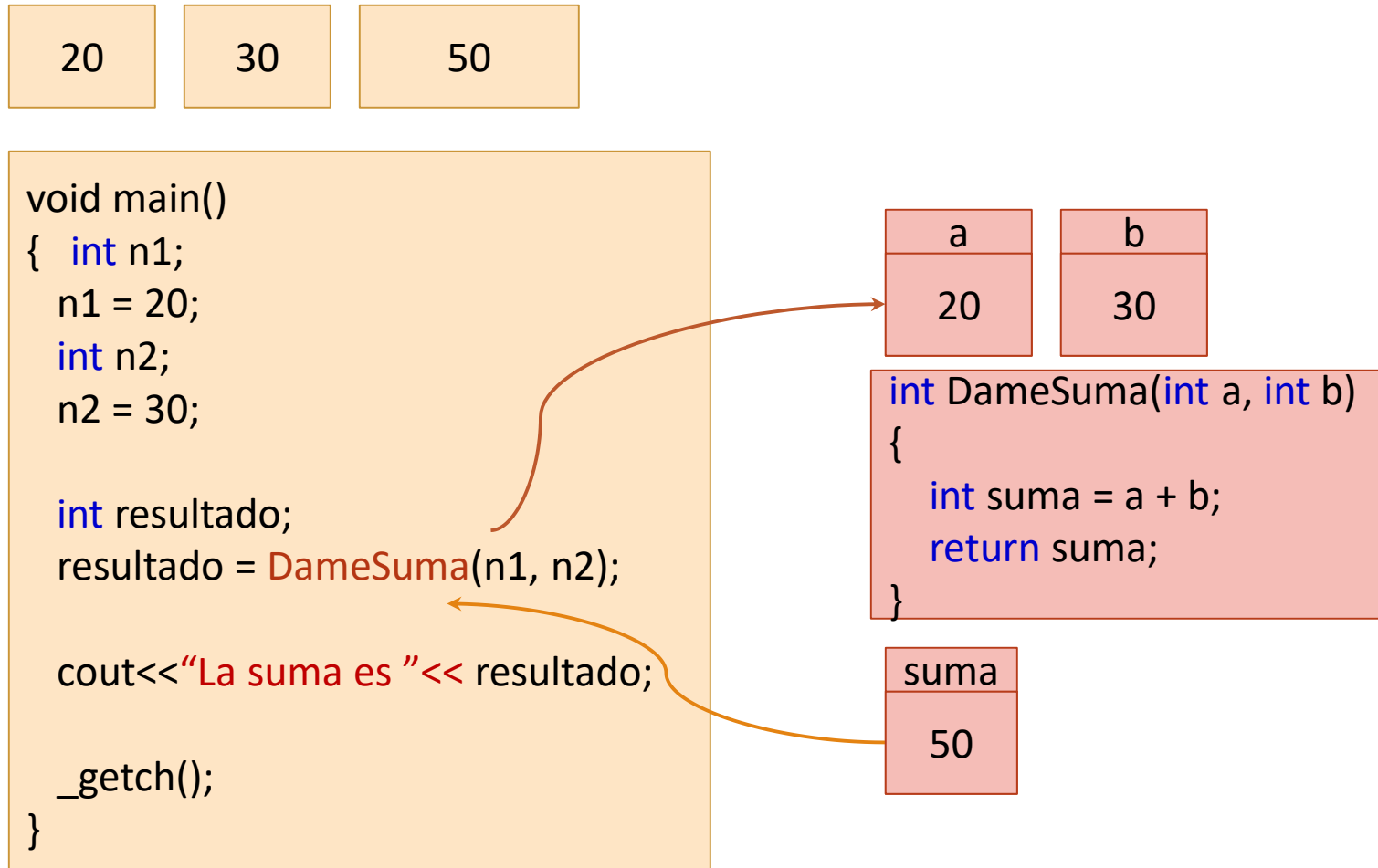
Con  
retorno

```
int DevuelveSuma()
{ int a,b,suma;
  cout <<"Ingrese 2 enteros:";
  cin >>a>>b;
  suma = a + b;
  return suma;
}
```

```
int DevuelveSuma(int a, int b)
{
  int suma = a + b;
  return suma;
}
```

## 1.3 FUNCION EN C++

### FUNCION CON PARAMETROS POR VALOR O COPIA



NOTA: Las variables en el main y los parámetros en DameSuma pueden llamarse igual, pero son diferentes.

# CONCLUSIONES

C++ es un lenguaje de programación orientado a objetos que toma la base del lenguaje C y le agrega la capacidad de usar mecanismos que permitieran la manipulación de objetos.

Los tipos de datos en C++ hace referencia al tipo de datos que puede contener y cómo almacena los datos en una variable, constante y otros. Los tipos de datos se aplican a todos los valores que se pueden almacenar en la memoria del equipo o participar en la evaluación de una expresión.

Una función en C++ es un fragmento de código que realiza una tarea bien definida al cual se le puede llamar por su nombre.

