



Universidad Politécnica Territorial
Agroindustrial del estado Táchira

Ing. Julio Alexander Zambrano Mendoza



Introducción A Lenguaje “C++” (SE LEE “ce plus plus”)

*“Por una educación universitaria
con calidad y pertenencia”*

📞 0276-3465260

🌐 www.uptaivirtualsarec.com

📺 📱 📷 📺 uptai_tachira



Universidad Politécnica Territorial
Agroindustrial del estado Táchira

Ing. Julio Alexander Zambrano Mendoza

ESTRUCTURA DE UN PROGRAMA EN "C++"

1-. Área de Mensajes
/* MENSAJE */

2-. Área de Librerías
include <nombre_librería>

3-. Área de Funciones
"Declaración de Funciones"

4-. Área de Variables Globales

5-. Función Principal
int main ()

6-. Área de Variables Locales

7-. Área de Definición de
Funciones

*"Por una educación universitaria
con calidad y pertenencia"*

0276-3465260



www.uptaivirtualesarec.com

uptai_tachira



ESTRUCTURA DE UN PROGRAMA EN "C++"

Área de Mensajes

// Comentario de una sola línea

/* Comentario de varias líneas */

Área de Librerías

#include <nombre_librería>

Área de Declaración de Funciones

Se declara dependiendo del tipo

Área de Variables Globales

Área de la Función Principal

int main()

{

Área de Variables Locales

}

Área de Desarrollo de Funciones





DECLARACIÓN DE VARIABLES SEGÚN EL TIPO DE DATOS

TIPO DE DATO	DESCRIPCIÓN
int	<i>Número Entero</i>
float	<i>Número con decimales</i>
double	<i>Número con decimales más largos</i>
char	<i>Carácter Simple</i>
string	<i>Cadena de caracteres</i>
bool	<i>Verdadero o Falso</i>



OPERADORES ARITMÉTICOS

SÍMBOLO	DESCRIPCIÓN
+	Suma
-	Resta
*	Multiplicación
/	División
%	Módulo (Residuo)
=	Asignación



OPERADORES COMPUESTOS DE ASIGNACIÓN

OPERADOR	PROPÓSITO	EJEMPLO	SALIDA
+=	Suma/asignación	X = 10; X += 5;	15
-=	Resta/asignación	X = 10; X -= 5;	5
*=	Multiplicación/asignación	X = 10; X *= 5;	50
/=	División/asignación	X = 10; X /= 5;	2
%=	Modulo/asignación	X = 10; X %= 5;	0



OPERADORES LÓGICOS

SÍMBOLO	DESCRIPCIÓN
&&	Y – and – Conjunción
	O – or – Disyunción
!	No – not – Negación



OPERADORES RELACIONALES

SÍMBOLO	DESCRIPCIÓN
<	Mayor que
>	Menor que
>=	Mayor e igual que
<=	Menor e igual que
!=	Diferente que
==	Igual que



Universidad Politécnica Territorial
Agroindustrial del estado Táchira

Ing. Julio Alexander Zambrano Mendoza

OPERADORES DE INCREMENTO

SÍMBOLO	DESCRIPCIÓN
$k++$ $++k$	Incremento
$k--$ $--k$	Decremento

“Por una educación universitaria
con calidad y pertenencia”

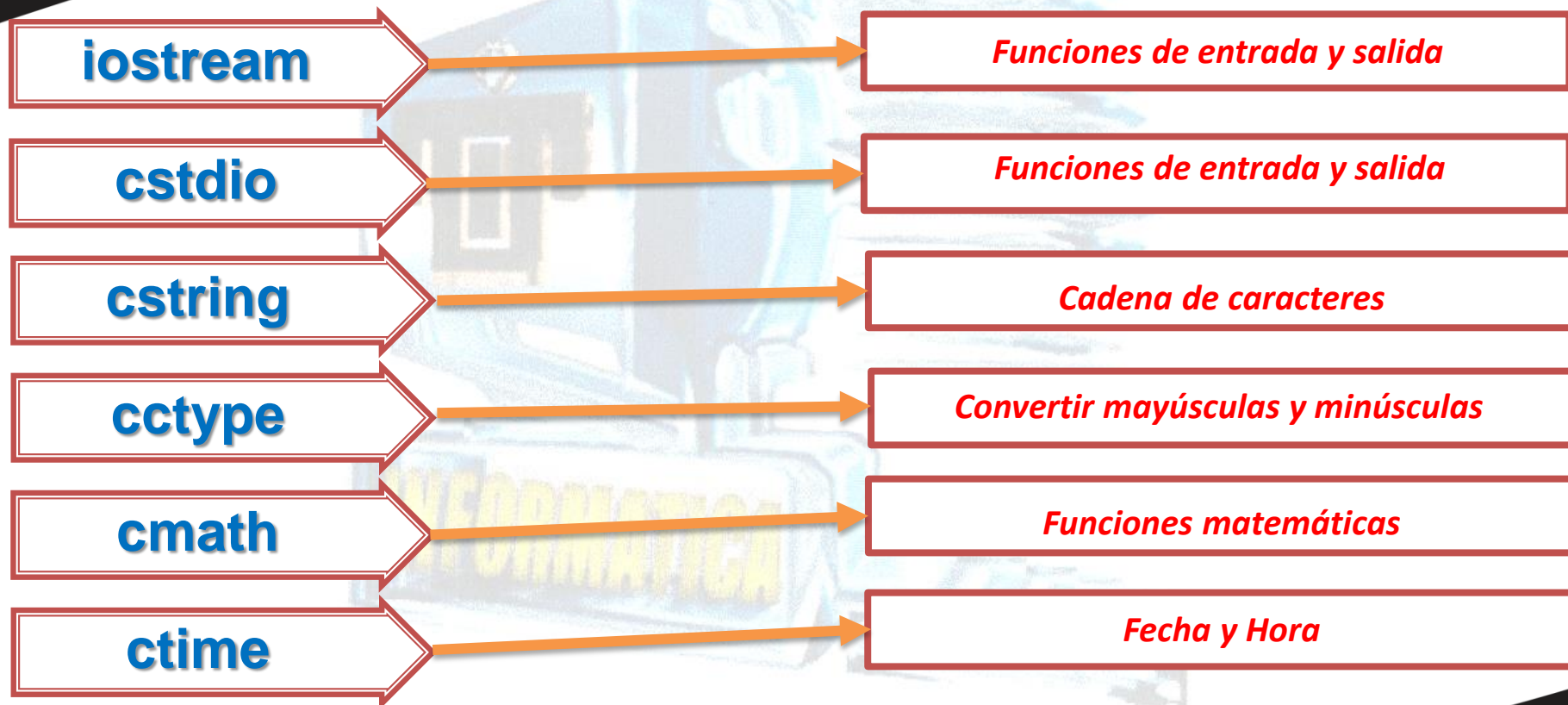
0276-3465260

 www.uptaivirtualsarec.com

    [uptai_tachira](https://www.instagram.com/uptai_tachira)



LIBRERIAS O BIBLIOTECAS





LIBRERIAS O BIBLIOTECAS





Universidad Politécnica Territorial
Agroindustrial del estado Táchira

Ing. Julio Alexander Zambrano Mendoza

SINTAXIS PARA LLAMADOS A LIBRERÍAS

SE UTILIZA:

include <nombre de la librería>

Se recomienda al final del llamado a librerías usar:

using namespace std;

En caso de que no se declare el uso del namespace std cada vez que se quiera usar cout, se tendría que escribir:

std::cout << "Hola mundo";

**“Por una educación universitaria
con calidad y pertenencia”**

 0276-3465260

 www.uptaivirtualsarec.com

    uptai_tachira



SENTENCIAS DE E/S

ENTRADA:

cin



Sintaxis:

cin >> variable;



SENTENCIAS DE E/S

SALIDA:

cout

Sintaxis:

Mensajes por pantalla:

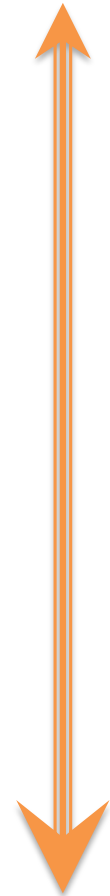
```
cout << "MENSAJE" << endl;
```

Mostrar contenido de variables:

```
cout << variable << endl;
```

Mensaje y contenido de Variable:

```
cout << "MENSAJE" << variable << endl;
```





Universidad Politécnica Territorial
Agroindustrial del estado Táchira

Ing. Julio Alexander Zambrano Mendoza

Funciones en Lenguaje “C++”

*“Por una educación universitaria
con calidad y pertenencia”*

 0276-3465260

 www.uptaivirtualsarec.com

    uptai_tachira



Es un conjunto secuencial de líneas de código de un lenguaje de programación, que ejecuta una tarea específica y puede retornar un valor al programa original.





COMPONENTES DE UNA FUNCIÓN

Parámetros

Valores que recibe la función como datos de entrada.

Código

Las sentencias de código de programación que internamente hace la función.

Resultado

Valor de retorno o valor final que entrega la función.



TIPOS DE FUNCIONES

Aquellas cuyo tipo de valor de retorno es
void (nulo)

```
int suma(int x, int y);  
{  
    int a;  
    a = x + y;  
    return(a);  
}
```

```
void main( )  
{  
    int x;  
    ....  
    x = potencia(a,b);  
    ...  
}
```

Aquellas cuyo tipo de valor de retorno en
un tipo de dato (int, double, float, char).



EJERCICIOS Nº 1

Desarrollar un algoritmo con la aplicación de una función que no recibe ni devuelve ningún parámetro. El algoritmo debe mostrar como salida el siguiente mensaje: “Hola Mundo”.



EJEMPLO DE FUNCIONES CON USO DEL VOID()

```
/* ESTA FUNCION NO RECIBE NINGUN VALOR */  
#include <iostream>  
using namespace std;  
// ÁREA DE DECLARACIÓN DE FUNCIONES  
void saludar();  
// proceso principal, que invoca a las funciones antes declaradas  
int main()  
{  
    cout << "Llamada a la funcion Saludar:" << endl<< endl<< endl;  
    saludar();  
    return 0;  
}  
// AREA DE DESARROLLO DE FUNCIONES  
void saludar()  
{  
    cout << "H o l a   . . .   m u n d o   !" << endl;  
}
```

Desarrollar un algoritmo con la aplicación de una función por valor, que recibe como parámetro una variable numérica entera y retorna el valor doble del parámetro recibido.



EJEMPLO DE FUNCIONES CON UN TIPO DE DATO()

```
/* ESTA FUNCIÓN RECIBE UN PARAMETRO Y RETORNA EL DOBLE SIN PERDER EL VALOR ORIGINAL RECIBIDO */  
# include <iostream>  
using namespace std;  
// AREA DE DECLARACION DE FUNCIONES  
int CalcularDoble(int num);  
// AREA VARIABLES GLOBALES  
int x;  
// AREA DE LA FUNCIÓN PRINCIPAL  
int main()  
{  
    cout << "Ingrese un valor numerico para x:" << endl;  
    cin >> x;  
    cout << "Llamada a la función CalcularDoble (pasaje por valor)" << endl << endl << endl;  
    cout << "El doble de " << x << " es ";  
    cout << CalcularDoble(x);  
    return 0;  
}  
// AREA DE DESARROLLO DE FUNCIONES  
int CalcularDoble(int num)  
{  
    int res=0;  
    res = num*2;  
    return res;  
}
```





Desarrollar un programa que permita leer 2 dígitos enteros y posteriormente mostrar en pantalla los números leídos.



// AREA DE MENSAJES

/* 3) Desarrollar un algoritmo/programa que permita leer 2 dígitos enteros posteriormente mostrar en pantalla los números registrados */

// AREA DE LIBRERIAS

```
# include <iostream>
# include <cstdlib>
# include <cstdio>
using namespace std;
```

// AREA DE VARIABLES GLOBALES

```
int dig1, dig2;
```

*“Por una educación universitaria
con calidad y pertenencia”*



// AREA DE MENSAJES

/* 1) Desarrollar un algoritmo/programa que permita leer 2 dígitos enteros posteriormente mostrar en pantalla los números registrados */

// AREA DE LIBRERIAS

```
# include <iostream>
# include <cstdlib>
# include <cstdio>
using namespace std;
```

// AREA DE VARIABLES GLOBALES

```
int dig1, dig2;
```



// AREA DE MENSAJES

/* 1) Desarrollar un algoritmo/programa que permita leer 2 dígitos enteros posteriormente mostrar en pantalla los números registrados */

// AREA DE LIBRERIAS

```
# include <iostream>
# include <cstdlib>
# include <cstdio>
using namespace std;
```

// AREA DE VARIABLES GLOBALES

```
int dig1, dig2;
```



```
cout << " " << endl << endl << endl;
cout << "    ... FINALIZO EL REGISTRO DE DATOS ... " << endl << endl;
cout << "... Presione la tecla < ENTER > para continuar ... " << endl;
getchar();
getchar();
system ("clear");
cout << "===== " << endl;
cout << "  DIGITOS REGISTRADOS " << endl;
cout << "===== " << endl;
cout << " " << endl;
cout << "  Digito Nro. 1: " << dig1 << endl;
cout << " " << endl;
cout << "  Digito Nro. 2: " << dig2 << endl;
cout << " " << endl;
cout << "===== " << endl;
return 0;
}
```

REALIZAR ESTE MISMO EJERCICIO PERO AHORA CON APLICACIÓN DE FUNCIONES

*"Por una educación universitaria
con calidad y pertenencia"*