



**Grado en Ingeniería Información**

# **PROGRAMACIÓN I**

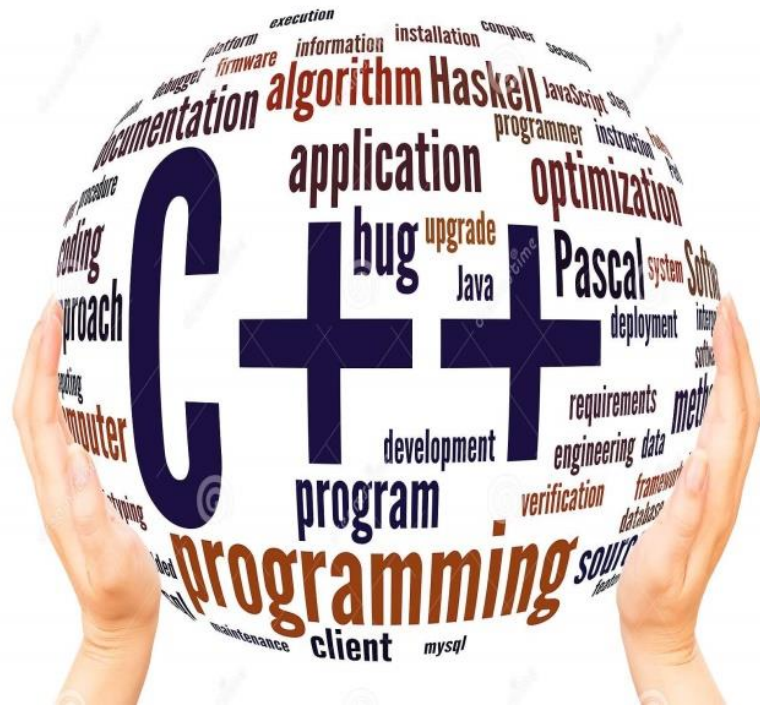
**Sesión 8**

**Curso 2022-2023**

Marta N. Gómez (mgomezper@nebrija.es)

# Funciones

- Funciones sin parámetros.
- Funciones con parámetros.
  - Parámetros por valor.
  - Parámetros por referencia.



ÍNDICE

# Funciones

- Las **funciones** son partes de código (**bloques de código**) que **realizan una tarea** al que se le asigna un **nombre (etiqueta)**.
- **Ámbito de la función** es todo lo que **existe dentro** del **bloque de código de la función**.
- Permiten **descomponer** el programa en **bloques de código (tareas)** que **se llaman entre ellos**.
- Siempre tiene que existir la función **main** dentro del código del programa. Esta la función es la que **arranca la ejecución del programa**.

# Funciones

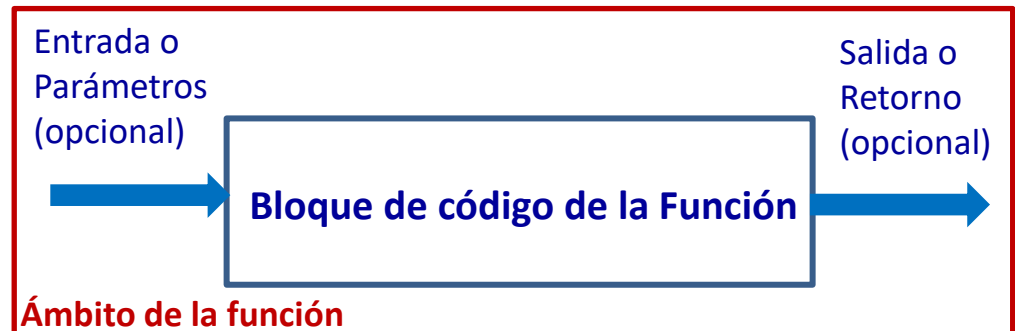
- **Sintaxis de la declaración o prototipo** de la **función** que es la **parte externa** o pública (conocida):

***tipoSalida** **nombreFuncion** (**listaParametrosFormalesEntrada**);*

- **Sintaxis de la definición** de la **función** que es la **parte interna** o privada (oculta):

***tipoSalida** **nombreFuncion** (**listaParametrosFormalesEntrada**)*

```
{ // Aquí se incluye el código de la función  
  // y la declaración de las variables locales  
  // que se necesiten  
}
```



# Funciones

```
#include <iostream>
```

```
using namespace std;
```

```
void saludar ();
```

**Prototipo de la Función**

```
int main()
```

```
{
```

```
    cout << "\n\n\tAquí empiezan las FUNCIONES";
```

```
    saludar();
```

**Llamada a la Función**

```
    cout << "\n\n\t";
```

```
    return 0;
```

```
}
```

**Definición de la Función**

```
void saludar ()
```

```
{
```

```
    cout << "\n\n\tHola, soy la FUNCION saludar.";
```

```
}
```

# Funciones – Ámbito de la función

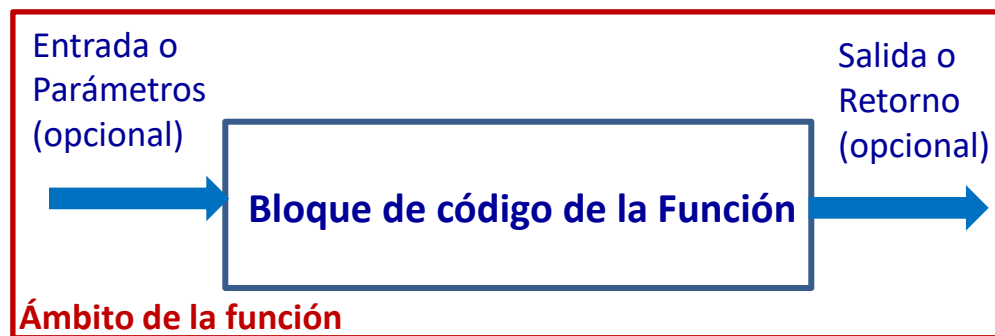
Las funciones, dentro de su **ámbito**, tienen como **datos**:

- **Variables locales**

- ✓ Se declaran en el cuerpo de la función.
- ✓ Sólo serán **conocidas dentro de la función**.
- ✓ Se **crean** y se **destruyen** con la función.

- **Parámetros**

- ✓ Permiten la **comunicación** de la función con el resto del programa mediante la transferencia de datos.
- ✓ Pueden ser: por **valor** o por **referencia**.



- **Funciones sin parámetros.**
- Funciones con parámetros.
  - Parámetros por valor.
  - Parámetros por referencia.



# ÍNDICE

# Funciones sin parámetros

- Son las funciones más básicas.
- Las funciones **SIN** parámetros **NO** reciben datos de entrada.
- Retorno o salida es **opcional**.
- **Sintaxis:**

***tipoSalida nombreFuncion ()***

***{ // Código de la función y las variables locales necesarias.***

***// expresión es la salida de la función***

***// y su tipo debe coincidir con tipoSalida.***

***// si tipoSalida es void se puede evitar el return***

***[return (expresión);]***

***}***



# Funciones sin parámetros y sin valor de retorno

```
#include <iostream>

using namespace std;
```

```
void saludar ();
```

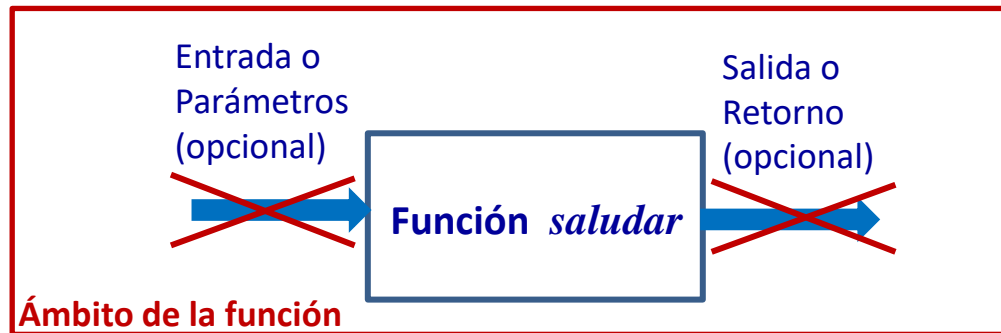
```
int main()
{
    cout << "\n\n\tAqui empiezan las FUNCIONES";

    saludar();

    cout << "\n\n\t";
    return 0;
}
```

```
void saludar ()
```

```
{
    cout << "\n\n\tHola, soy la FUNCION saludar.";
}
```



```
Aqui empiezan las FUNCIONES

Hola, soy la FUNCION saludar.

Press <RETURN> to close this window...
```

# Funciones sin parámetros con retorno simple

```
#include <iostream>
```

```
using namespace std;
```

```
int pedirDato ();
```

```
int main()
```

```
{
```

```
    cout << "\n\n\tFUNCION CON RETORNO SIMPLE";
```

```
    int numero = pedirDato();
```

```
    cout << "\n\n\tEl numero introducido es: " << numero << "\n\n\t";
```

```
    return 0;
```

```
}
```

```
int pedirDato ()
```

```
{    int dato;
```

```
    cout << "\n\n\tTeclee un numero: ";
```

```
    cin >> dato;
```

```
    return dato;
```

```
}
```

FUNCION CON RETORNO SIMPLE

Teclee un numero: 25

El numero introducido es: 25

Press <RETURN> to close this window...

Entrada o  
Parámetros  
(opcional)

Ámbito de la función



# Funciones sin parámetros con retorno de *struct*

```
#include <iostream>
#include <string>

using namespace std;

struct Persona
{
    string nombre, apellido;
    int edad;
};

Persona pedirDato ();

int main()
{
    cout << "\n\n\tFUNCION CON RETORNO STRUCT";

    Persona amigo = pedirDato();
    cout << "\n\n\tMi amigo se llama: " << amigo.nombre << " |"
        << amigo.apellido << " y su edad es " << amigo.edad;

    cout << "\n\n\t";
    return 0;
}
```

```
Persona pedirDato ()
{
    Persona una_persona;

    cout << "\n\n\tIntroduzca su nombre: ";
    getline (cin, una_persona.nombre);
    cout << "\n\n\tIntroduzca su apellido: ";
    getline (cin, una_persona.apellido);
    cout << "\n\n\tIntroduzca su edad: ";
    cin >> una_persona.edad;

    return una_persona;
}
```

FUNCION CON RETORNO STRUCT

Introduzca su nombre: Alberto

Introduzca su apellido: Martinez

Introduzca su edad: 25

Mi amigo se llama: Alberto Martinez y su edad es 25

Press <RETURN> to close this window...

Entrada o  
Parámetros  
(opcional)

Ámbito de la función



# Funciones sin parámetros con retorno *array*

```
#include <iostream>
#include <array>

using namespace std;

array<char,5> pedirDatos ();

int main()
{
    cout << "\n\n\tFUNCION CON RETORNO ARRAY";

    array <char, 5> vocales = pedirDatos();

    cout << "\n\n\tLas vocales son: ";
    for(char letra:vocales)
        cout << letra << " ";
    cout << "\n\n\t";
    return 0;
}
```

```
array<char,5> pedirDatos ()
{
    array <char, 5> caracteres;

    cout << "\n\n\tIntroduzca las vocales: \n";
    for(unsigned int i{0}; i < caracteres.size(); i++)
    {
        cout << "\t\t\t\t";
        cin >> caracteres.at(i);
    }

    return caracteres;
}
```

## FUNCION CON RETORNO ARRAY

Introduzca las vocales:

a  
e  
i  
o  
u

Las vocales son: a e i o u

Press <RETURN> to close this window...

Entrada o  
Parámetros  
(opcional)

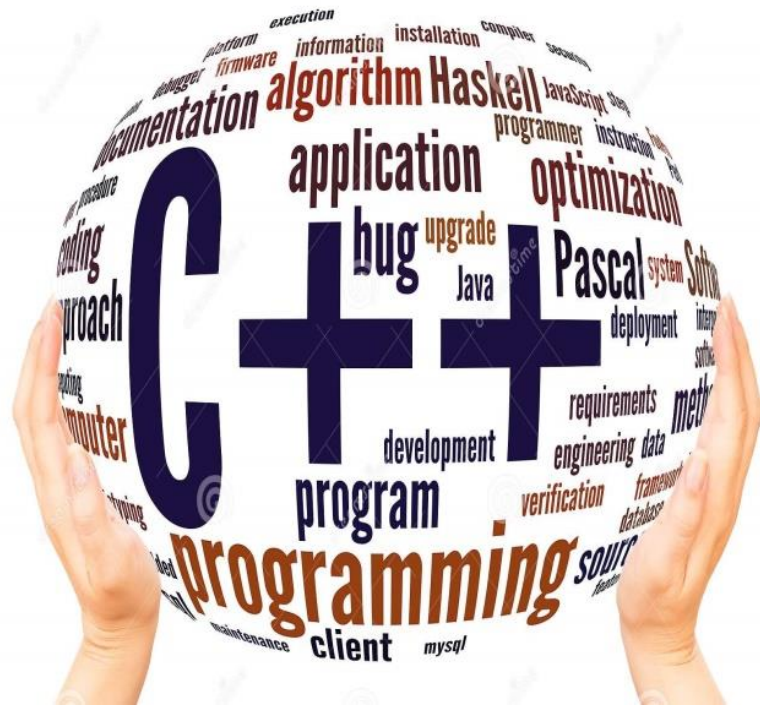
Ámbito de la función

array

Función *pedirDatos*

# Funciones

- Funciones sin parámetros.
- **Funciones con parámetros.**
  - Parámetros por valor.
  - Parámetros por referencia.



# Funciones con parámetros

La función **recibe datos de entrada** a través de los **parámetros** que permiten **interactuar** con el resto del programa.

## – Paso de parámetros por valor

Los parámetros de la función (*parámetros formales*) son **una copia** de los valores enviados en la llamada a la función (*parámetros reales o actuales*).

## – Paso de parámetros por referencia

Los parámetros de la función (*parámetros formales*) reciben **la dirección de memoria de la variable original** enviada en la llamada a la función.

# Funciones con parámetros

- Sintaxis:

**tipoSalida** **nombreFuncion** (*listaParametrosFormalesEntrada*)

{ // Código de la función y las variables locales necesarias.

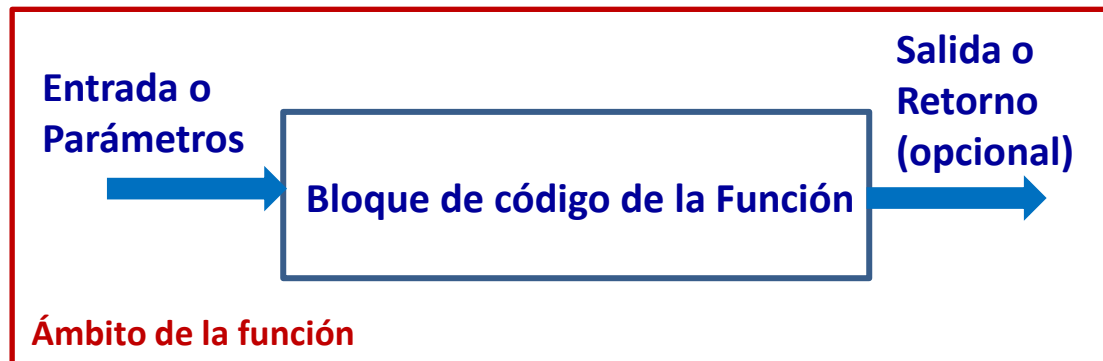
// **expresión** es la salida de la función

// y su **tipo** debe coincidir con **tipoSalida**

// si **tipoSalida** es **void** se puede evitar el **return**

**[return (expresión);]**

}



# Funciones

- Funciones sin parámetros.
- **Funciones con parámetros.**
  - **Parámetros por valor.**
  - Parámetros por referencia.





## Funciones con parámetros - Paso por valor (por copia)

- Los parámetros pueden ser de cualquier tipo.
- Los **parámetros de la función** son **una copia** de los **valores enviados** en la llamada a la función.
- El compilador **reserva memoria** para los **parámetros formales**, donde almacena una **copia del valor enviado**.
- El proceso “**gasta**” recursos de **tiempo** y **memoria** (espacio).
- Si los valores originales se envían a través de **variables**, **la función no puede modificar sus contenidos**:

Los cambios efectuados sobre el **parámetro formal** dentro de la función ***no se mantienen*** en el **parámetro real o actual** (fuera del ámbito de la función).

# Funciones con parámetros - Paso por valor (por copia)

```
#include <iostream>

using namespace std;

int suma(int a, int b);

int main() {
    int num1, num2;
    cout << "\n\n\tIntroduzca el primer sumando: ";
    cin >> num1;
    cout << "\n\n\tIntroduzca el segundo sumando: ";
    cin >> num2;

    int resultado = suma(num1, num2);

    cout << "\n\n\tLa suma de " << num1 << " + " << num2 << " = " << resultado;

    cout << "\n\n\t";
    return 0;
}

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

Parámetros  
Formales

Introduzca el primer sumando: 4

Introduzca el segundo sumando: 6

La suma de 4 + 6 = 10

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por valor (por copia)

```
#include <iostream>
```

```
using namespace std;
```

```
int suma(int a, int b);
```

```
int main() {
```

```
    int num1, num2;
```

```
    cout << "\n\n\tIntroduzca el primer sumando: ";
```

```
    cin >> num1;
```

```
    cout << "\n\n\tIntroduzca el segundo sumando: ";
```

```
    cin >> num2;
```

```
    int resultado = suma(num1, num2);
```

```
    cout << "\n\n\tLa suma de " << num1 << " + " << num2 << " = " << resultado;
```

```
    cout << "\n\n\t";
```

```
    return 0;
```

```
}
```

```
int suma(int a, int b)
```

```
{
```

```
    return a+b;
```

```
}
```

num1	num2			resultado
4	6	a	b	

Introduzca el primer sumando: 4

Introduzca el segundo sumando: 6

La suma de 4 + 6 = 10

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por valor (por copia)

```
#include <iostream>
```

```
using namespace std;
```

```
int suma(int a, int b);
```

```
int main() {
```

```
    int num1, num2;
```

```
    cout << "\n\n\tIntroduzca el primer sumando: ";
```

```
    cin >> num1;
```

```
    cout << "\n\n\tIntroduzca el segundo sumando: ";
```

```
    cin >> num2;
```

```
    int resultado = suma(num1, num2);
```

```
    cout << "\n\n\tLa suma de " << num1 << " + " << num2 << " = " << resultado;
```

```
    cout << "\n\n\t";
```

```
    return 0;
```

```
}
```

```
int suma(int a, int b)
```

```
{
```

```
    return a+b;
```

```
}
```

num1	num2	resultado	
4	6	a	b

Parámetros  
Actuales

Introduzca el primer sumando: 4

Introduzca el segundo sumando: 6

La suma de 4 + 6 = 10

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por valor (por copia)

```
#include <iostream>
```

```
using namespace std;
```

```
int suma(int a, int b);
```

```
int main() {
```

```
    int num1, num2;
```

```
    cout << "\n\n\tIntroduzca el primer sumando: ";
```

```
    cin >> num1;
```

```
    cout << "\n\n\tIntroduzca el segundo sumando: ";
```

```
    cin >> num2;
```

```
    int resultado = suma(num1, num2);
```

```
    cout << "\n\n\tLa suma de " << num1 << " + " << num2 << " = " << resultado;
```

```
    cout << "\n\n\t";
```

```
    return 0;
```

```
}
```

```
int suma(int a, int b)
```

```
{
```

```
    return a+b;
```

```
}
```

num1	num2	resultado	
4	6	a	b
		4	6

Introduzca el primer sumando: 4

Introduzca el segundo sumando: 6

La suma de 4 + 6 = 10

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por valor (por copia)

```
#include <iostream>
```

```
using namespace std;
```

```
int suma(int a, int b);
```

```
int main() {
```

```
    int num1, num2;
```

```
    cout << "\n\n\tIntroduzca el primer sumando: ";
```

```
    cin >> num1;
```

```
    cout << "\n\n\tIntroduzca el segundo sumando: ";
```

```
    cin >> num2;
```

```
    int resultado = suma(num1, num2);
```

```
    cout << "\n\n\tLa suma de " << num1 << " + " << num2 << " = " << resultado;
```

```
    cout << "\n\n\t";
```

```
    return 0;
```

```
}
```

```
int suma(int a, int b)
```

```
{
```

```
    return a+b;
```

```
}
```

num1	num2	resultado	
4	6	a	b
		4	6
			10

```
Introduzca el primer sumando: 4
```

```
Introduzca el segundo sumando: 6
```

```
La suma de 4 + 6 = 10
```

```
Press <RETURN> to close this window...
```

# Funciones con parámetros - Paso por valor (por copia)

```
#include <iostream>
```

```
using namespace std;
```

```
int suma(int a, int b);
```

```
int main() {
```

```
    int num1, num2;
```

```
    cout << "\n\n\tIntroduzca el primer sumando: ";
```

```
    cin >> num1;
```

```
    cout << "\n\n\tIntroduzca el segundo sumando: ";
```

```
    cin >> num2;
```

```
    int resultado = suma(num1, num2);
```

```
    cout << "\n\n\tLa suma de " << num1 << " + " << num2 << " = " << resultado;
```

```
    cout << "\n\n\t";
```

```
    return 0;
```

```
}
```

```
int suma(int a, int b)
```

```
{
```

```
    return a+b;
```

```
}
```

num1	num2	resultado	
4	6	a	b
		4	6
			10

Introduzca el primer sumando: 4

Introduzca el segundo sumando: 6

La suma de 4 + 6 = 10

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por valor (por copia)

```
#include <iostream>

using namespace std;

void intercambio(int a, int b);

int main() {
    int num1, num2;
    cout << "\n\n\tIntroduzca el primer valor: ";
    cin >> num1;
    cout << "\n\n\tIntroduzca el segundo valor: ";
    cin >> num2;

    cout << "\n\n\tAntes del intercambio: num1 = " << num1 << " num2 = " << num2;

    intercambio(num1, num2);

    cout << "\n\n\tDespues del intercambio: num1 = " << num1 << " num2 = " << num2;

    cout << "\n\n\t";
    return 0;
}

void intercambio(int a, int b)
{ int c[a];
  cout << "\n\n\tAl entrar de la funcion: a = " << a << " b = " << b;
  a = b;
  b = c;

  cout << "\n\n\tAl salir de la funcion: a = " << a << " b = " << b;
}
```

num1	num2	a	b	c
4	6			

Introduzca el primer valor: 4

Introduzca el segundo valor: 6

Antes del intercambio: num1 = 4 num2 = 6

Al entrar de la funcion: a = 4 b = 6

Al salir de la funcion: a = 6 b = 4

Despues del intercambio: num1 = 4 num2 = 6

Press <RETURN> to close this window...



# Funciones con parámetros - Paso por valor (por copia)

```
#include <iostream>

using namespace std;

void intercambio(int a, int b);

int main() {
    int num1, num2;
    cout << "\n\n\tIntroduzca el primer valor: ";
    cin >> num1;
    cout << "\n\n\tIntroduzca el segundo valor: ";
    cin >> num2;

    cout << "\n\n\tAntes del intercambio: num1 = " << num1 << " num2 = " << num2;

    intercambio(num1, num2);

    cout << "\n\n\tDespues del intercambio: num1 = " << num1 << " num2 = " << num2;

    cout << "\n\n\t";
    return 0;
}

void intercambio(int a, int b)
{ int c[a];
  cout << "\n\n\tAl entrar de la funcion: a = " << a << " b = " << b;
  a = b;
  b = c;

  cout << "\n\n\tAl salir de la funcion: a = " << a << " b = " << b;
}
```

num1	num2	a	b	c
4	6			

Introduzca el primer valor: 4

Introduzca el segundo valor: 6

Antes del intercambio: num1 = 4 num2 = 6

Al entrar de la funcion: a = 4 b = 6

Al salir de la funcion: a = 6 b = 4

Despues del intercambio: num1 = 4 num2 = 6

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por valor (por copia)

```
#include <iostream>

using namespace std;

void intercambio(int a, int b);

int main() {
    int num1, num2;
    cout << "\n\n\tIntroduzca el primer valor: ";
    cin >> num1;
    cout << "\n\n\tIntroduzca el segundo valor: ";
    cin >> num2;

    cout << "\n\n\tAntes del intercambio: num1 = " << num1 << " num2 = " << num2;

    intercambio(num1, num2);

    cout << "\n\n\tDespues del intercambio: num1 = " << num1 << " num2 = " << num2;

    cout << "\n\n\t";
    return 0;
}

void intercambio(int a, int b)
{ int c[a];
  cout << "\n\n\tAl entrar de la funcion: a = " << a << " b = " << b << "\n";
  a = b;
  b = c;

  cout << "\n\n\tAl salir de la funcion: a = " << a << " b = " << b << "\n";
}
```

num1	num2	a	b	c
4	6			
		4	6	4

Introduzca el primer valor: 4

Introduzca el segundo valor: 6

Antes del intercambio: num1 = 4 num2 = 6

Al entrar de la funcion: a = 4 b = 6

Al salir de la funcion: a = 6 b = 4

Despues del intercambio: num1 = 4 num2 = 6

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por valor (por copia)

```
#include <iostream>

using namespace std;

void intercambio(int a, int b);

int main() {
    int num1, num2;
    cout << "\n\n\tIntroduzca el primer valor: ";
    cin >> num1;
    cout << "\n\n\tIntroduzca el segundo valor: ";
    cin >> num2;

    cout << "\n\n\tAntes del intercambio: num1 = " << num1 << " num2 = " << num2;

    intercambio(num1, num2);

    cout << "\n\n\tDespues del intercambio: num1 = " << num1 << " num2 = " << num2;

    cout << "\n\n\t";
    return 0;
}

void intercambio(int a, int b)
{ int c[a];
  cout << "\n\n\tAl entrar de la funcion: a = " << a << " b = " << b;
  a = b;
  b = c;

  cout << "\n\n\tAl salir de la funcion: a = " << a << " b = " << b;
}
```

num1	num2	a	b	c
4	6			
		6	4	4

Introduzca el primer valor: 4

Introduzca el segundo valor: 6

Antes del intercambio: num1 = 4 num2 = 6

Al entrar de la funcion: a = 4 b = 6

Al salir de la funcion: a = 6 b = 4

Despues del intercambio: num1 = 4 num2 = 6

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por valor (por copia)

```
#include <iostream>

using namespace std;

void intercambio(int a, int b);

int main() {
    int num1, num2;
    cout << "\n\n\tIntroduzca el primer valor: ";
    cin >> num1;
    cout << "\n\n\tIntroduzca el segundo valor: ";
    cin >> num2;

    cout << "\n\n\tAntes del intercambio: num1 = " << num1 << " num2 = " << num2;

    intercambio(num1, num2);

    cout << "\n\n\tDespues del intercambio: num1 = " << num1 << " num2 = " << num2;

    cout << "\n\n\t";
    return 0;
}

void intercambio(int a, int b)
{ int c[a];
  cout << "\n\n\tAl entrar de la funcion: a = " << a << " b = " << b << "\n";
  a = b;
  b = c;

  cout << "\n\n\tAl salir de la funcion: a = " << a << " b = " << b << "\n";
}
```

num1	num2	a	b	c
4	6			

Introduzca el primer valor: 4

Introduzca el segundo valor: 6

Antes del intercambio: num1 = 4 num2 = 6

Al entrar de la funcion: a = 4 b = 6

Al salir de la funcion: a = 6 b = 4

Despues del intercambio: num1 = 4 num2 = 6

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por valor (por copia)

```
#include <iostream>

using namespace std;

int calcularTriple(int num);

int main() {
    int triple, numero{20};

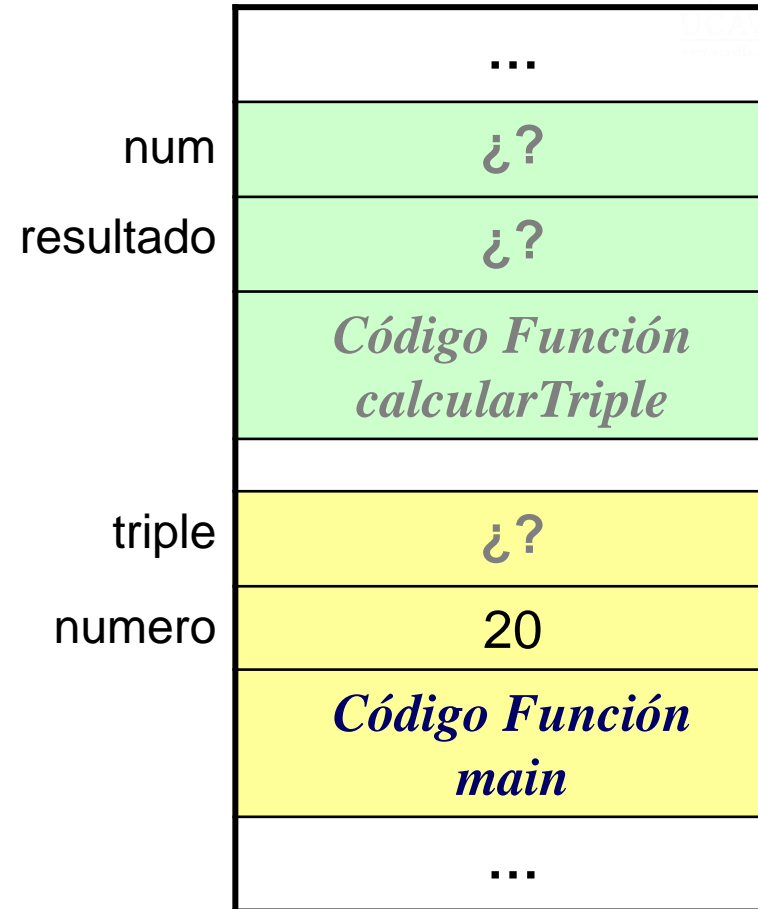
    triple = calcularTriple(numero);

    cout << "\n\n\tEl triple de " << numero
          << " es " << triple;

    cout << "\n\n\t";
    return 0;
}

int calcularTriple (int num)
{
    int resultado = 3 * num;

    return resultado;
}
```



**MEMORIA**

El triple de 20 es 60

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por valor (por copia)

```
#include <iostream>

using namespace std;

int calcularTriple(int num);

int main() {
    int triple, numero{20};

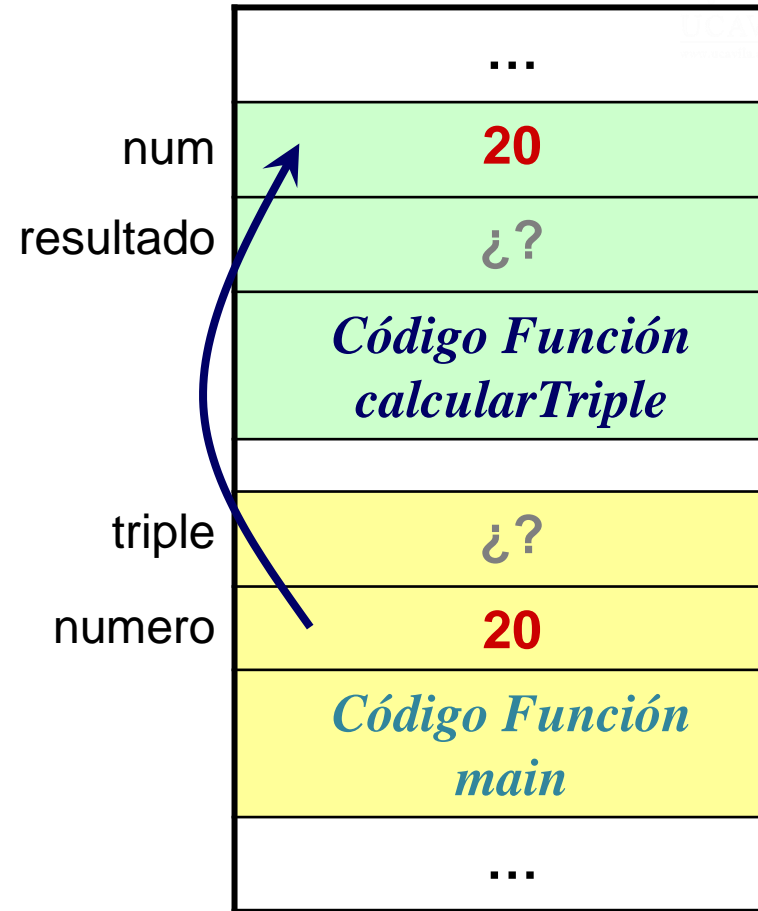
    triple = calcularTriple(numero);

    cout << "\n\n\tEl triple de " << numero
         << " es " << triple;

    cout << "\n\n\t";
    return 0;
}

int calcularTriple (int num)
{
    int resultado = 3 * num;

    return resultado;
}
```



**MEMORIA**

El triple de 20 es 60

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por valor (por copia)

```
#include <iostream>

using namespace std;

int calcularTriple(int num);

int main() {
    int triple, numero{20};

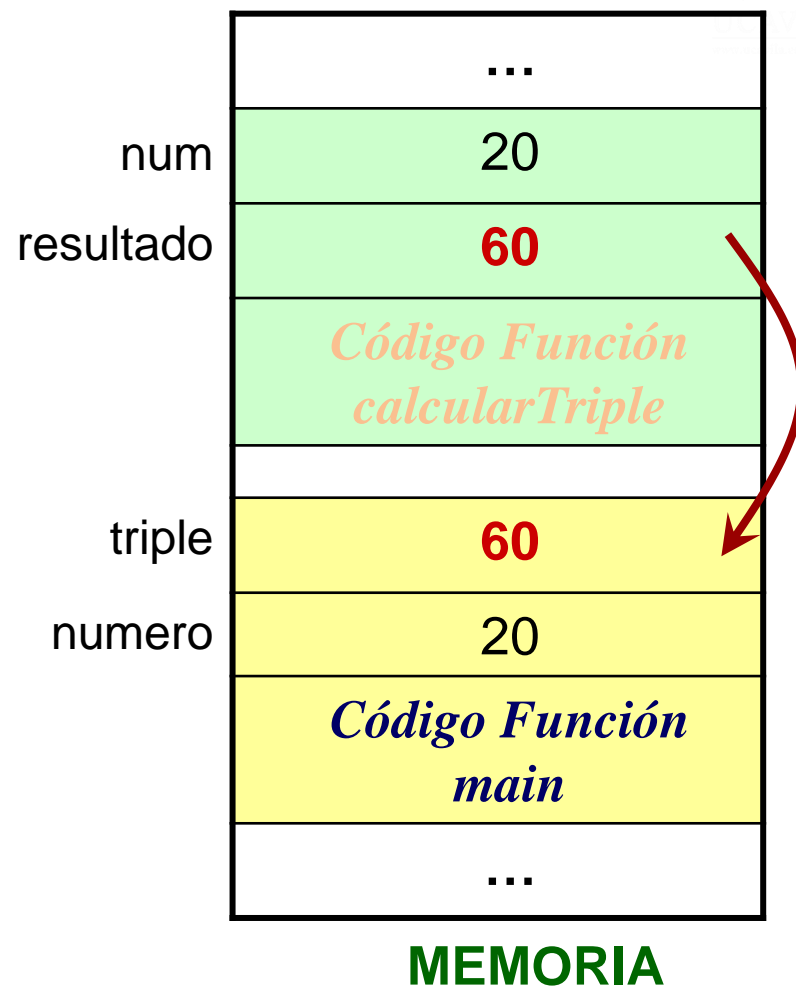
    triple = calcularTriple(numero);

    cout << "\n\n\tEl triple de " << numero
          << " es " << triple;

    cout << "\n\n\t";
    return 0;
}

int calcularTriple (int num)
{
    int resultado = 3 * num;

    return resultado;
}
```



```
El triple de 20 es 60
```

```
Press <RETURN> to close this window...
```



**Grado en Ingeniería Información**

# **PROGRAMACIÓN I**

**Sesión 9**

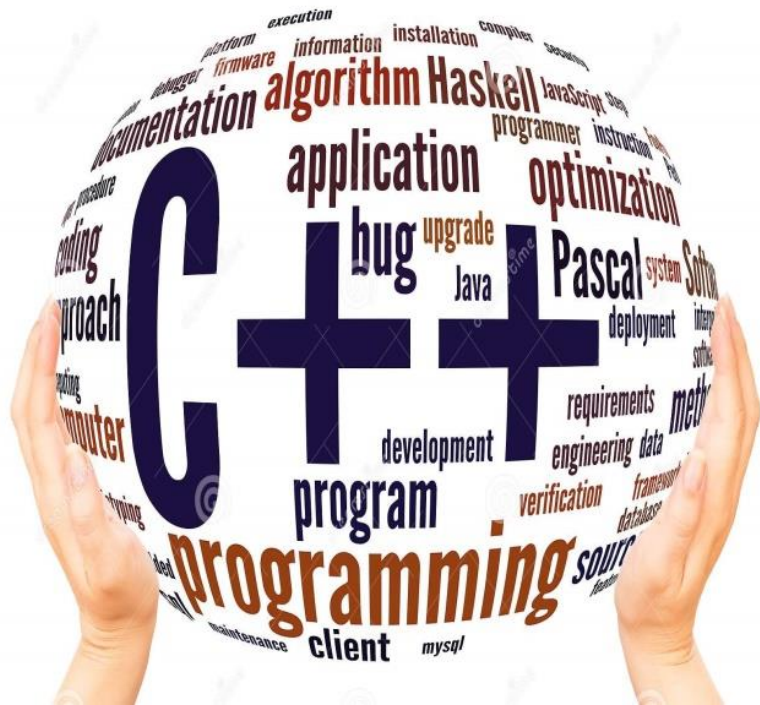
**Curso 2022-2023**

Marta N. Gómez (mgomezper@nebrija.es)



# Funciones

- Funciones sin parámetros.
- **Funciones con parámetros.**
  - Parámetros por valor.
  - **Parámetros por referencia.**



# ÍNDICE

# Funciones con parámetros - Paso por referencia

- Los parámetros pueden ser de cualquier tipo.
- Los parámetros de la función (*parámetros formales*) reciben **la dirección de memoria de la variable** original enviada en la llamada a la función.
- Los cambios efectuados sobre el **parámetro formal** dentro de la función **se mantienen** en el **parámetro real o actual** (fuera del ámbito de la función).

**tipoSalida** **nombreFuncion** (**tipo\_p1** &p1, **tipo\_p2** &p2, ...)

{ // Código de la función y las variables

// locales necesarias.

[**return** (expresión);]

}

operador dirección

# Funciones con parámetros - Paso por referencia incorrecto

```
#include <iostream>
```

```
using namespace std;
```

```
int suma(int &a, int &b);
```

```
int main() {
```

```
    int num1, num2;
```

```
    cout << "\n\n\tIntroduzca el primer sumando: ";
```

```
    cin >> num1;
```

```
    cout << "\n\n\tIntroduzca el segundo sumando: ";
```

```
    cin >> num2;
```

```
    int resultado = suma(num1, num2);
```

```
    cout << "\n\n\tLa suma de " << num1 << " + " << num2 << " = " << resultado;
```

```
    cout << "\n\n\t";
```

```
    return 0;
```

```
}
```

```
int suma(int &a, int &b)
```

```
{
```

```
    return a+b;
```

```
}
```

num1	num2	resultado	
5	6	a	b
<d1087>	<d1207>		

Introduzca el primer sumando: 5

Introduzca el segundo sumando: 6

La suma de 5 + 6 = 11

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por referencia incorrecto

```
#include <iostream>
```

```
using namespace std;
```

```
int suma(int &a, int &b);
```

```
int main() {
```

```
    int num1, num2;
```

```
    cout << "\n\n\tIntroduzca el primer sumando: ";
```

```
    cin >> num1;
```

```
    cout << "\n\n\tIntroduzca el segundo sumando: ";
```

```
    cin >> num2;
```

```
    int resultado = suma(num1, num2);
```

```
    cout << "\n\n\tLa suma de " << num1 << " + " << num2 << " = " << resultado;
```

```
    cout << "\n\n\t";
```

```
    return 0;
```

```
}
```

```
int suma(int &a, int &b)
```

```
{
```

```
    return a+b;
```

```
}
```

num1	num2	resultado		
5	6	a	b	
<d1087>	<d1207>	<d1087>	<d1207>	

Introduzca el primer sumando: 5

Introduzca el segundo sumando: 6

La suma de 5 + 6 = 11

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por referencia incorrecto

```
#include <iostream>
```

```
using namespace std;
```

```
int suma(int &a, int &b);
```

```
int main() {
```

```
    int num1, num2;
```

```
    cout << "\n\n\tIntroduzca el primer sumando: ";
```

```
    cin >> num1;
```

```
    cout << "\n\n\tIntroduzca el segundo sumando: ";
```

```
    cin >> num2;
```

```
    int resultado = suma(num1, num2);
```

```
    cout << "\n\n\tLa suma de " << num1 << " + " << num2 << " = " << resultado;
```

```
    cout << "\n\n\t";
```

```
    return 0;
```

```
}
```

```
int suma(int &a, int &b)
```

```
{
```

```
    return a+b;
```

```
}
```

num1	num2	resultado		
5	6	a	b	11
<d1087>	<d1207>	<d1087>	<d1207>	<d1110>

Introduzca el primer sumando: 5

Introduzca el segundo sumando: 6

La suma de 5 + 6 = 11

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por referencia

```
#include <iostream>

using namespace std;

void intercambio(int &a, int &b)

int main() {
    int num1, num2;
    cout << "\n\n\tIntroduzca el primer valor: ";
    cin >> num1;
    cout << "\n\n\tIntroduzca el segundo valor: ";
    cin >> num2;
```

```
    cout << "\n\n\tAntes del intercambio: num1 = " << num1 << " num2 = " << num2;

    intercambio(num1, num2);

    cout << "\n\n\tDespues del intercambio: num1 = " << num1 << " num2 = " << num2;

    cout << "\n\n\t";
    return 0;
}
```

```
void intercambio(int &a, int &b)
{ int c[a];
    cout << "\n\n\tAl entrar de la funcion: a = " << a << " b = " << b;
    a = b;
    b = c;

    cout << "\n\n\tAl salir de la funcion: a = " << a << " b = " << b;
}
```

num1	num2	a	b	c
4	6			
<d1873>	<d2038>			

Introduzca el primer valor: 4

Introduzca el segundo valor: 6

Antes del intercambio: num1 = 4 num2 = 6

Al entrar de la funcion: a = 4 b = 6

Al salir de la funcion: a = 6 b = 4

Despues del intercambio: num1 = 6 num2 = 4

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por referencia

```
#include <iostream>
```

```
using namespace std;
```

```
void intercambio(int &a, int &b);
```

```
int main() {
```

```
    int num1, num2;
```

```
    cout << "\n\n\tIntroduzca el primer valor: ";
```

```
    cin >> num1;
```

```
    cout << "\n\n\tIntroduzca el segundo valor: ";
```

```
    cin >> num2;
```

```
    cout << "\n\n\tAntes del intercambio: num1 = " << num1 << " num2 = " << num2;
```

```
    intercambio(num1, num2);
```

```
    cout << "\n\n\tDespues del intercambio: num1 = " << num1 << " num2 = " << num2;
```

```
    cout << "\n\n\t";
```

```
    return 0;
```

```
}
```

```
void intercambio(int &a, int &b)
```

```
{ int c[a];
```

```
    cout << "\n\n\tAl entrar de la funcion: a = " << a << " b = " <<
```

```
    a = b;
```

```
    b = c;
```

```
    cout << "\n\n\tAl salir de la funcion: a = " << a << " b = " <<
```

```
}
```

num1	num2	a	b	c
4	6			
<d1873>	<d2038>			

Introduzca el primer valor: 4

Introduzca el segundo valor: 6

Antes del intercambio: num1 = 4 num2 = 6

Al entrar de la funcion: a = 4 b = 6

Al salir de la funcion: a = 6 b = 4

Despues del intercambio: num1 = 6 num2 = 4

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por referencia

```
#include <iostream>
```

```
using namespace std;
```

```
void intercambio(int &a, int &b);
```

```
int main() {
```

```
    int num1, num2;
```

```
    cout << "\n\n\tIntroduzca el primer valor: ";
```

```
    cin >> num1;
```

```
    cout << "\n\n\tIntroduzca el segundo valor: ";
```

```
    cin >> num2;
```

```
    cout << "\n\n\tAntes del intercambio: num1 = " << num1 << " num2 = " << num2;
```

```
    intercambio(num1, num2);
```

```
    cout << "\n\n\tDespues del intercambio: num1 = " << num1 << " num2 = " << num2;
```

```
    cout << "\n\n\t";
```

```
    return 0;
```

```
}
```

```
void intercambio(int &a, int &b)
```

```
{ int c[a];
```

```
    cout << "\n\n\tAl entrar de la funcion: a = " << a << " b = " <<
```

```
    a = b;
```

```
    b = c;
```

```
    cout << "\n\n\tAl salir de la funcion: a = " << a << " b = " <<
```

```
}
```

num1	num2	a	b	c
4	6			
<d1873>	<d2038>	<d1873>	<d2038>	4

Introduzca el primer valor: 4

Introduzca el segundo valor: 6

Antes del intercambio: num1 = 4 num2 = 6

Al entrar de la funcion: a = 4 b = 6

Al salir de la funcion: a = 6 b = 4

Despues del intercambio: num1 = 6 num2 = 4

Press <RETURN> to close this window...



# Funciones con parámetros - Paso por referencia

```
#include <iostream>

using namespace std;

void intercambio(int &a, int &b);

int main() {
    int num1, num2;
    cout << "\n\n\tIntroduzca el primer valor: ";
    cin >> num1;
    cout << "\n\n\tIntroduzca el segundo valor: ";
    cin >> num2;
```

```
    cout << "\n\n\tAntes del intercambio: num1 = " << num1 << " num2 = " << num2;
```

```
    intercambio(num1, num2);
```

```
    cout << "\n\n\tDespues del intercambio: num1 = " << num1 << " num2 = " << num2;
```

```
    cout << "\n\n\t";
    return 0;
}
```

```
void intercambio(int &a, int &b)
```

```
{ int c[a];
```

```
    cout << "\n\n\tAl entrar de la funcion: a = " << a << " b = " <<
```

```
    a = b;
```

```
    b = c;
```

```
    cout << "\n\n\tAl salir de la funcion: a = " << a << " b = " <<
```

num1	num2	a	b	c
6	4			
<d1873>	<d2038>	<d1873>	<d2038>	4

Introduzca el primer valor: 4

Introduzca el segundo valor: 6

Antes del intercambio: num1 = 4 num2 = 6

Al entrar de la funcion: a = 4 b = 6

Al salir de la funcion: a = 6 b = 4

Despues del intercambio: num1 = 6 num2 = 4

Press <RETURN> to close this window...

# Funciones con parámetros - Paso por referencia

```
#include <iostream>
```

```
using namespace std;
```

```
void intercambio(int &a, int &b);
```

```
int main() {
```

```
    int num1, num2;
```

```
    cout << "\n\n\tIntroduzca el primer valor: ";
```

```
    cin >> num1;
```

```
    cout << "\n\n\tIntroduzca el segundo valor: ";
```

```
    cin >> num2;
```

```
    cout << "\n\n\tAntes del intercambio: num1 = " << num1 << " num2 = " << num2;
```

```
    intercambio(num1, num2);
```

```
    cout << "\n\n\tDespues del intercambio: num1 = " << num1 << " num2 = " << num2;
```

```
    cout << "\n\n\t";
```

```
    return 0;
```

```
}
```

```
void intercambio(int &a, int &b)
```

```
{ int c[a];
```

```
    cout << "\n\n\tAl entrar de la funcion: a = " << a << " b = " <<
```

```
    a = b;
```

```
    b = c;
```

```
    cout << "\n\n\tAl salir de la funcion: a = " << a << " b = " <<
```

```
}
```

num1	num2	a	b	c
6	4			
<d1873>	<d2038>	<d1873>	<d2038>	4

Introduzca el primer valor: 4

Introduzca el segundo valor: 6

Antes del intercambio: num1 = 4 num2 = 6

Al entrar de la funcion: a = 4 b = 6

Al salir de la funcion: a = 6 b = 4

Despues del intercambio: num1 = 6 num2 = 4

Press <RETURN> to close this window...

# Funciones

- Principios de la Programación Funcional:
  - Una **función recibe datos del exterior** (fuera de su ámbito) a través de **sus parámetros (entrada del exterior)**.
  - Una función realiza operaciones con los **valores recibidos sin modificar las variables externas**.
  - Una función proporciona datos al exterior a través del **valor de retorno (salida al exterior)**.
- Recomendaciones y buenas prácticas:
  - Una función **NO** debe **acceder a variables fuera de su ámbito**.
  - Una función **NO** debe **modificar variables fuera de su ámbito**.

# Funciones – Uso del Paso por Referencia

- Los **objetivos/motivos** del **paso de valores por referencia** son:
  - **Ahorro de memoria:** No duplicar los datos.
  - **Ahorro de tiempo:** No tener que hacer la copia de los datos.
- Los **tipos simples** (int, float, double, char, etc.) **siempre se pasarán por copia.**
- Los **tipos complejos** (struct, std::string, std::vector, etc.) **siempre se pasarán por referencia.**

# Funciones – Uso del Paso por Referencia

- Los parámetros pasados por referencia **NO** siempre se deben de **modificar**.
- Para evitar modificaciones inadecuadas/indeseadas, **siempre se pasarán** como **parámetro como referencia constante**.

```

tipoSalida nombreFuncion (tipo_p1 const &p1, tipo_p2 const &p2, ...)
{
    // Código de la función y las variables
    // locales necesarias.
    [return (expresión);]
}

```

Referencia constante

```

tipoSalida nombreFuncion (const tipo_p1 &p1, const tipo_p2 &p2, ...)
{
    // Código de la función y las variables
    // locales necesarias.
    [return (expresión);]
}

```

# F. con parámetros - Paso por referencia con efecto colateral

```
#include <iostream>

using namespace std;

void calcularTriple(int &num);

int main() {
    int numero{20};

    cout << "\n\n\tEl triple de " << numero;

    calcularTriple(numero);

    cout << " es " << numero;

    cout << "\n\n\t";
    return 0;
}
```

```
void calcularTriple (int &num)
{
    num = 3 * num;
}
```

	...	
num	<dir45>	<dir235>
	<i>Código Función calcularTriple</i>	
numero	20	<dir45>
	<i>Código Función main</i>	
	...	

**MEMORIA**

El triple de 20 es 60

Press <RETURN> to close this window...

# F. con parámetros - Paso por referencia con efecto colateral

```
#include <iostream>

using namespace std;

void calcularTriple(int &num);

int main() {
    int numero{20};

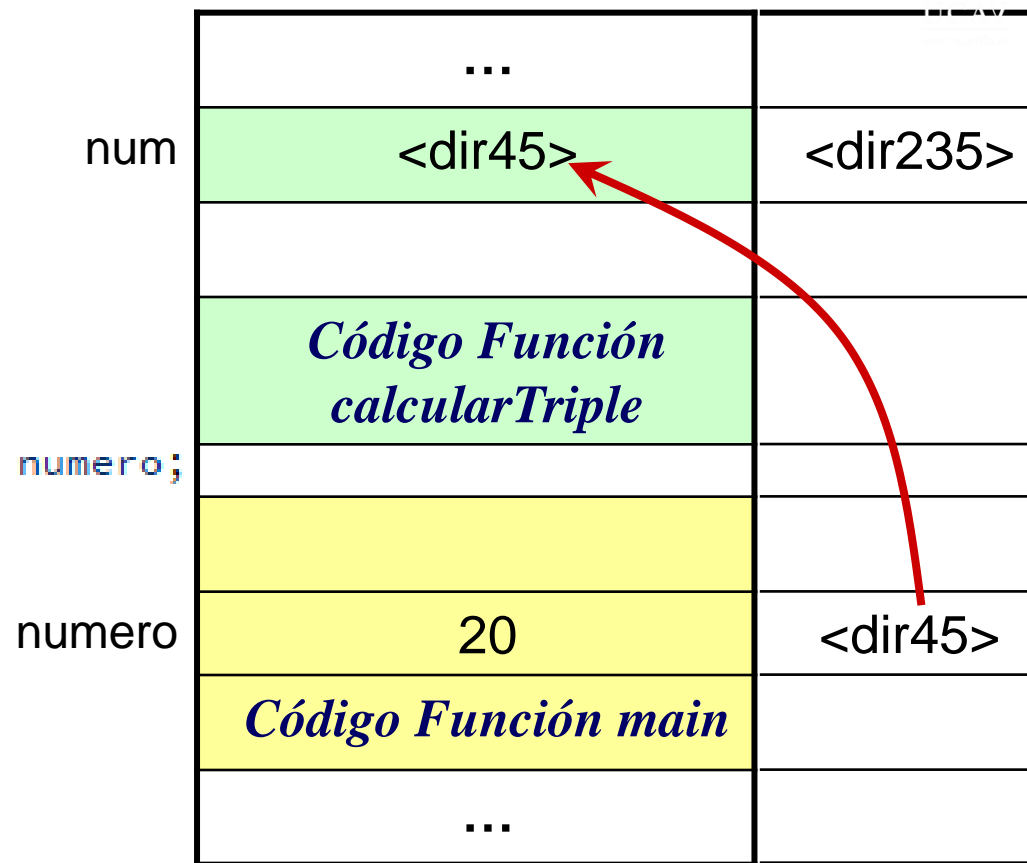
    cout << "\n\n\tEl triple de " << numero;

    calcularTriple(numero);

    cout << " es " << numero;

    cout << "\n\n\t";
    return 0;
}
```

```
void calcularTriple (int &num)
{
    num = 3 * num;
}
```



MEMORIA

El triple de 20 es 60

Press <RETURN> to close this window...

# F. con parámetros - Paso por referencia con efecto colateral

```
#include <iostream>

using namespace std;

void calcularTriple(int &num);

int main() {
    int numero{20};

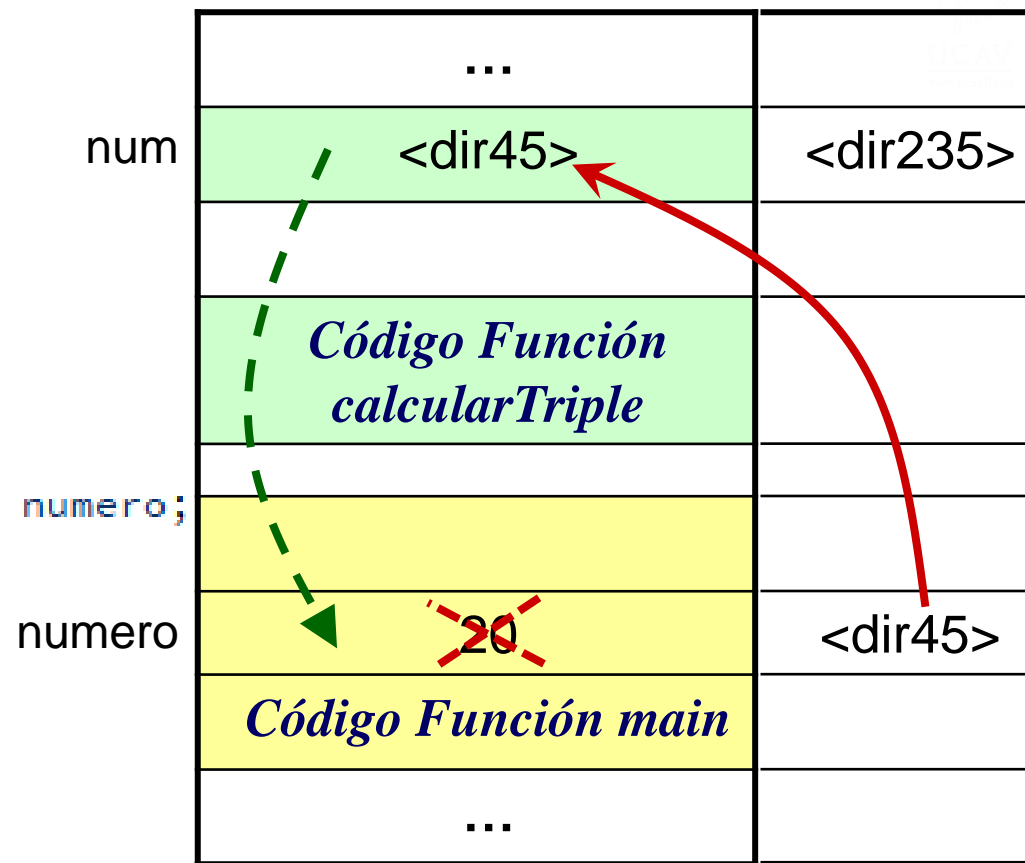
    cout << "\n\n\tEl triple de " << numero;

    calcularTriple(numero);

    cout << " es " << numero;

    cout << "\n\n\t";
    return 0;
}
```

```
void calcularTriple (int &num)
{
    num = 3 * num;
}
```



MEMORIA

El triple de 20 es 60

Press <RETURN> to close this window...



# F. con parámetros - Paso por referencia con efecto colateral

```
#include <iostream>

using namespace std;

void calcularTriple(int &num);

int main() {
    int numero{20};

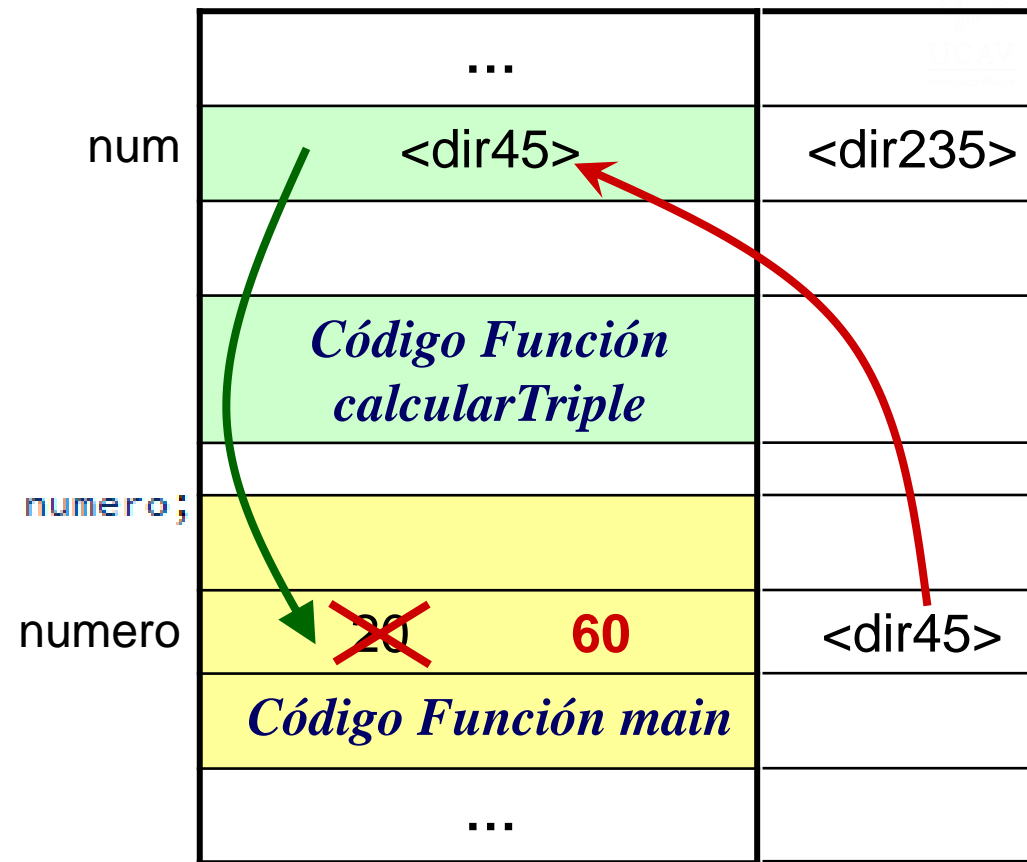
    cout << "\n\n\tEl triple de " << numero;

    calcularTriple(numero);

    cout << " es " << numero;

    cout << "\n\n\t";
    return 0;
}

void calcularTriple (int &num)
{
    num = 3 * num;
}
```



**MEMORIA**

El triple de 20 es 60

Press <RETURN> to close this window...

# F. con parámetros - Paso por referencia con efecto colateral

```
#include <iostream>

using namespace std;

void calcularTriple(int &num);

int main() {
    int numero{20};

    cout << "\n\n\tEl triple de " << numero;

    calcularTriple(numero);

    cout << " es " << numero;

    cout << "\n\n\t";
    return 0;
}

void calcularTriple (int &num)
{
    num = 3 * num;
}
```

num

resultado

numero

...	
<dir45>	<dir235>
60	<dir501>
<i>Código Función calcularTriple</i>	
60	<dir45>
<i>Código Función main</i>	
...	

**MEMORIA**

El triple de 20 es 60

Press <RETURN> to close this window...

```
#include <iostream>
#include <vector>

using namespace std;

void multiplicaPorDos(vector<int> & vectorN);

void mostrarVector(vector<int> const & vectorN);

int main() {
    vector<int> numeros{10,20,30,40};

    mostrarVector(numeros);

    multiplicaPorDos(numeros);

    mostrarVector(numeros);

    cout << "\n\n\t";
    return 0;
}

// Función para mostrar por pantalla el contenido del vector
void mostrarVector(vector<int> const & vectorN){

    cout << "\n\n\t";
    for(int elem: vectorN){
        cout << elem << "-";
    }
    cout << endl;
}

// Función que multiplica por dos el contenido del vector
void multiplicaPorDos(vector<int> & vectorN){

    for(int i{0}; i <= vectorN.size(); i++)
        vectorN.at(i) = vectorN.at(i) * 2;
}
```

**F. con parámetros - Paso por referencia sin efecto colateral**