



Modularización de programas:

Definición de módulo y características.

Paso de parámetros en los módulos.

Tipos de módulos: función y procedimiento

FUNDAMENTOS DE PROGRAMACIÓN



Teoría

Reconoce los módulos de programa, tipos de módulos tales como funciones y procedimientos en el lenguaje de programación.



Módulo de Programa – Paso de Parámetros

El módulo es
una porción
(segmento) de
programa

Paso de Parámetros

Ejemplo:

La variable "a"
**paso de parámetros
por valor.**

La variable "b"
**paso de parámetros
por referencia.**

```
1 #include<iostream>
2 using namespace std;
3
4 void modulo(int a, int &b)
5 {
6     a++;
7     b++;
8
9
10    cout<<"\n";
11    cout<<"El valor de a es: " << a;
12    cout<<"\n";
13    cout<<"El valor de b es: " << b;
14    cout<<"\n";
15 }
16
17 int main()
18 {
19     int x, y;
20
21     x=1;
22     y=1;
23
24     modulo(x,y);
25
26
27     cout<<"\n";
28     cout<<"El valor de x es: "<<x;
29     cout<<"\n";
30     cout<<"El valor de y es: "<<y;
31     cout<<"\n";
32
33     return 0;
34 }
```

int a

2

direccion de a

int &b

direccion de y

direccion de b

a es 2

b es 2

int x

1

direccion de x

int y

2

direccion de y

modulo(1,1)

x es 1

y es 2



Tipos de Módulo

La creación del módulo puede ser de dos formas:

a. Procedimiento

Realizan acciones de programa, pero **no devuelve** valor.

Declaración Sin argumentos:

void *NombreProcedimiento* ()

Declaración Con argumentos:

void *NombreProcedimiento* (**argumentos**)

b. Funciones

Realizan acciones de programa, y **devuelve** un valor.

Declaración Sin argumentos:

Tipo_dato *NombreFunción* ()

Declaración Con argumentos:

Tipo_dato *NombreFunción* (**argumentos**)

Declaración en la cabecera o prototipo:

```
1 #include<iostream>
2 using namespace std;
3
4 void nota (int); //Prototipo
5
6 int main()
7 {
8     int N;
9     cout<<"Ingrese una nota (0-20): ";
10    cin>>N;
11    nota (N); //Llamado
12    cout<<endl<<endl;
13    system("pause");
14    return 0;
15 }
16
17 void nota (int A) //Definición
18 {
19     if (A<0 || A>20)
20         cout<<"Nota incorrecta \n";
21     else
22         if (A<11)
23             cout<<"Desaprobado \n";
24         else
25             cout<<"Aprobado \n";
26 }
```

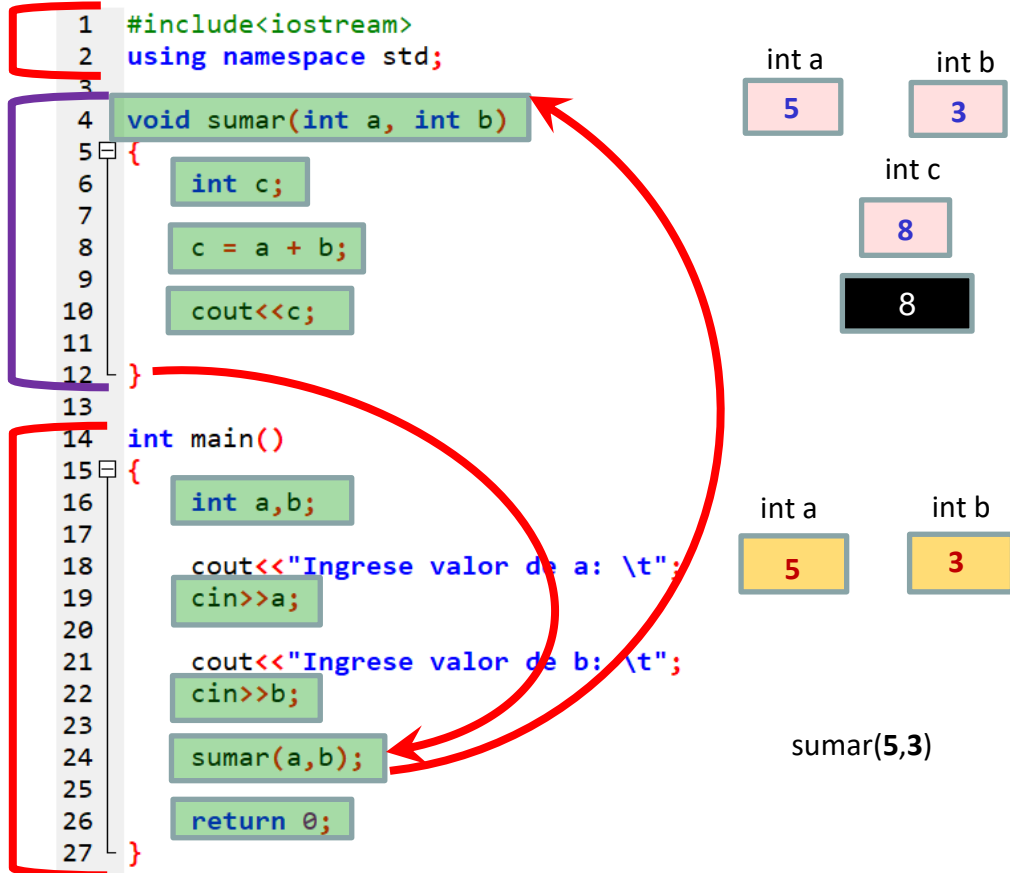
Antes del módulo que lo invoca:

```
1 #include<iostream>
2 using namespace std;
3
4 void nota (int A)
5 {
6     if (A<0 || A>20)
7         cout<<"Nota incorrecta \n";
8     else
9         if (A<11)
10             cout<<"Desaprobado \n";
11         else
12             cout<<"Aprobado \n";
13 }
14
15 int main()
16 {
17     int N;
18     cout<<"Ingrese una nota (0-20): ";
19     cin>>N;
20     nota (N); //Llamado
21     cout<<endl<<endl;
22     system("pause");
23     return 0;
24 }
```

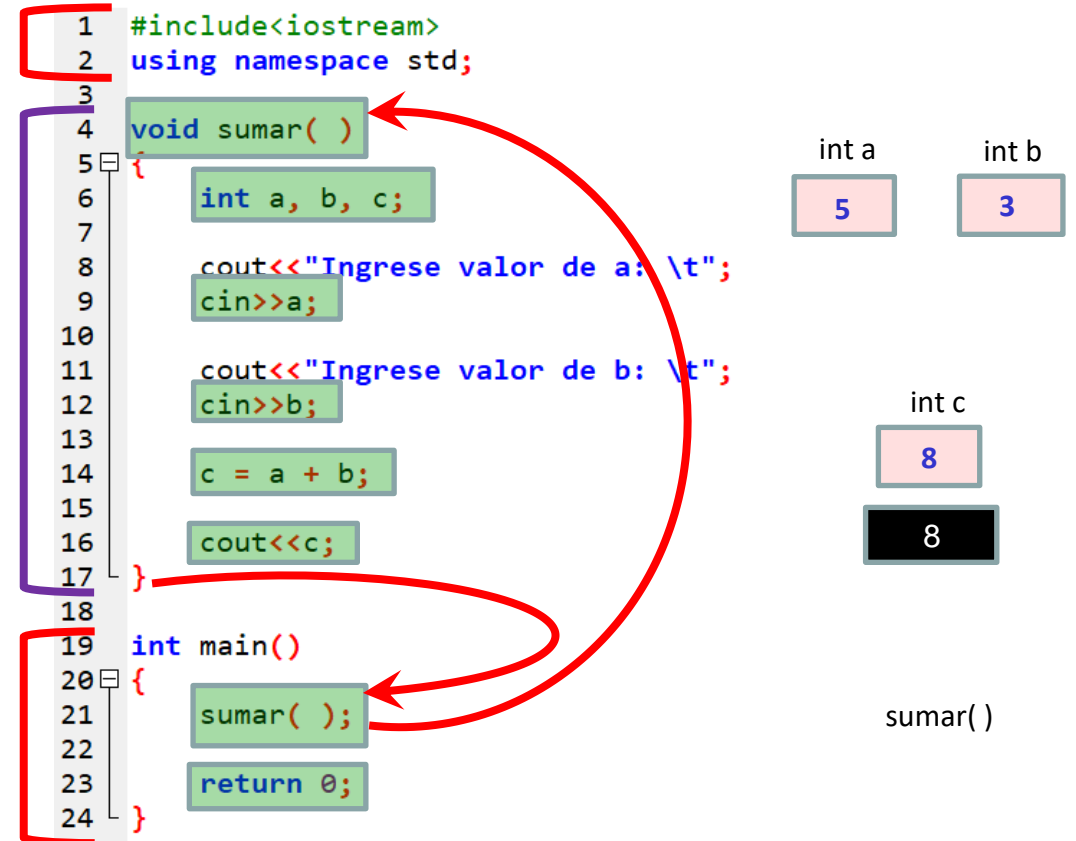


Procedimiento: no devuelve valores

Ejemplo **con** argumentos



Ejemplo **sin** argumentos

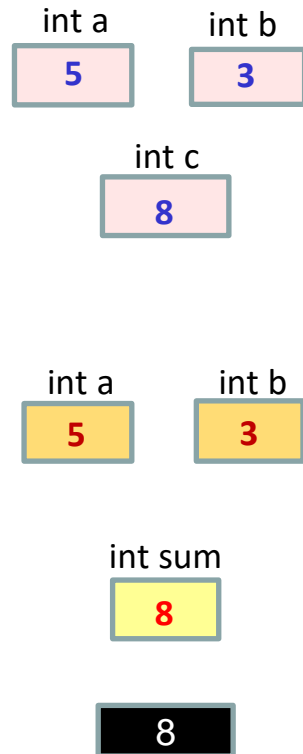




Función: devuelve un valor

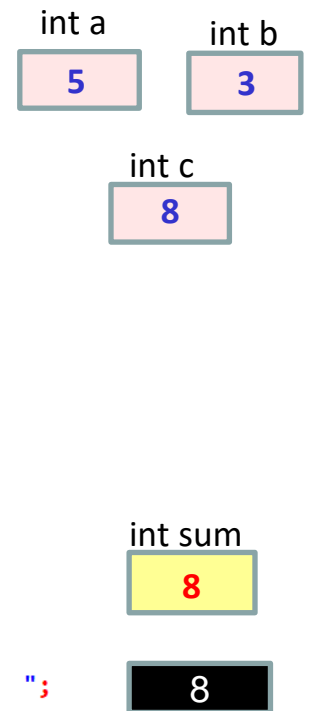
Ejemplo **con** argumentos

```
1 #include<iostream>
2 using namespace std;
3
4 int sumar(int a, int b)
5 {
6     int c;
7     c = a + b;
8     return c;
9 }
10
11
12
13 int main()
14 {
15     int a, b, sum;
16     cout<<"Ingrese valor de a: ";
17     cin>>a;
18     cout<<"Ingrese valor de b: ";
19     cin>>b;
20     sum = sumar(a,b);
21     cout<<"Valor de la suma de a y b: ";
22     cout<<sum;
23     return 0;
24 }
```



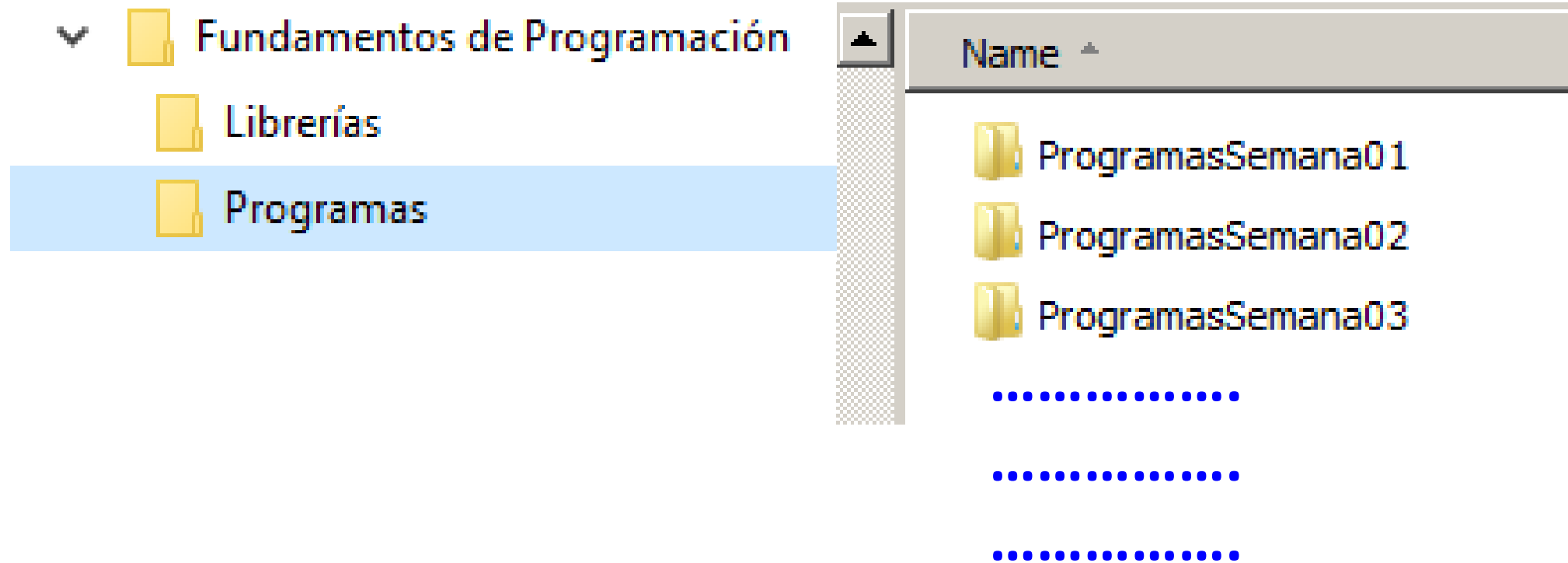
Ejemplo **sin** argumentos

```
1 #include<iostream>
2 using namespace std;
3
4 int sumar( )
5 {
6     int a,b,c;
7     cout<<"Ingrese valor de a: ";
8     cin>>a;
9     cout<<"Ingrese valor de b: ";
10    cin>>b;
11    c = a + b;
12    return c;
13 }
14
15
16
17 int main()
18 {
19     int sum;
20     sum = sumar( );
21     cout<<"Valor de la suma de a y b: ";
22     cout<<sum;
23     return 0;
24 }
```





Recomendación para crear los programas



Ahora, elaboremos los programas ejemplos..!



Ejemplos

EJEMPLO 1

```
1 #include<iostream>
2 using namespace std;
3
4 float ecuaciones()
5 {
6     float x, y;
7
8     cout<<"Ingrese valor de x: "<<endl;
9     cin>>x;;
10
11     if(x<=0)
12     {
13         cout<<"La funcion y = x^2 + 5 " <<endl;
14         y = (x*x) + 5;
15     }
16     else
17     {
18         if(x>0 && x<2)
19         {
20             cout<<"La funcion y = 3x - 1 ";
21             y = (3*x)-1;
22         }
23         else
24         {
25             if(x>=2)
26             {
27                 cout<<"La funcion y = x^2 - 4x + 5 " <<endl;
28                 y = (x*x) - (4*x)+5;
29             }
30         }
31     }
32     return y;
33 }
34
35 int main()
36 {
37     float ecuac;
38     ecuac = ecuaciones(); // invoca modulo ecuaciones sin argumentos
39     cout<<"Resultado de la funcion es: ";
40     cout<<ecuac<<endl;
41
42     return 0;
43 }
```

El módulo función `ecuaciones()` es invocado desde el principal `main()`, y luego retorna un valor.

EJEMPLO 2

```
8 #include<iostream>
9 using namespace std;
10
11 float leedatof()
12 {
13     float dato;
14     cin>>dato;
15     return dato;
16 }
17
18 float ecuaciones()
19 {
20     float x,y;
21
22     cout<<"Ingresar valor de x : \t";
23     x = leedatof(); // usando modulo de lectura float
24
25     if(x<=0)
26     {
27         cout<<endl<<"Ecuacion a usar es y=(x^2)+5\n";
28         y=(x*x)+5;
29     }
30     else
31     {
32         if(x>0&&x<2)
33         {
34             cout<<endl<<"Ecuacion a usar es y=3x+1\n";
35             y=(3*x)-1;
36         }
37         else
38         {
39             if(x>=2)
40             {
41                 cout<<endl<<"Ecuacion a usar es y=(x^2)-4x+5\n";
42                 y=(x*x)-(4*x)+5;
43             }
44         }
45     }
46     return y;
47 }
48
49 int main()
50 {
51     float res;
52     res=ecuaciones();
53
54     cout<<"Resultado de la funcion es f(y)= "<<res;
55     cout<<endl;
56
57     return 0;
58 }
59 }
```

Se implementa otro módulo función `leedatof()` para ingresar valores tipo float, y es invocado desde el módulo `ecuaciones()`, y luego retorna un valor.

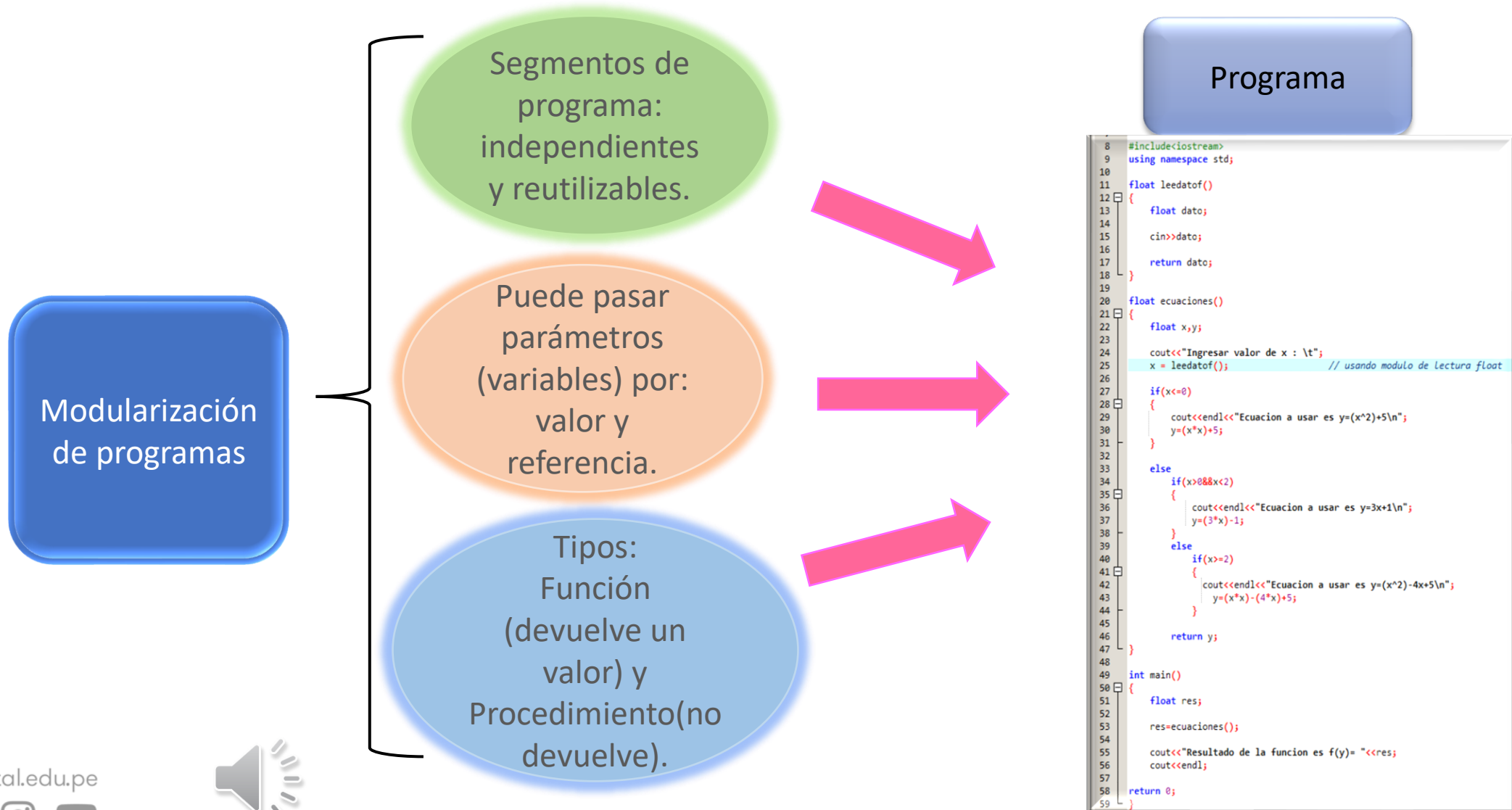
EJEMPLO 3

Cambiar `ecuaciones()` a procedimiento...!!!:

- Se puede cambiar?
- Que otros cambios se requiere para el programa ejecute correctamente?



Conclusiones





Práctica

Crea programas ejemplo, con paso de parámetros por valor y con los tipos de módulos, en el lenguaje de programación C/ C++.



Desarrollamos la Guía Práctica 09

Guía práctica N°09

Fundamentos de Programación

Tercera Unidad: Módulos para la programación: función y procedimiento

Sección :	Apellidos :
Docente :	Nombres :
	Fecha : / / Duración: 100 min
	Tipo de práctica: Individual (x) Grupal ()
Instrucciones: Lea detenidamente cada enunciado y desarrolle lo solicitado. Utilizar el Dev C++ para el desarrollo de los siguientes programas	

1. Elaborar un programa, que contenga un módulo Ventas() en la que se ingresa cantidad y precio de un producto y calcule el monto a pagar; y que sea invocado en el módulo principal. Puedes usar función o procedimiento, con o sin argumentos, según requiera el caso. Referencia de solución:

Carpeta: Programas Semana 05
Nombre Proyecto: Proyecto2_VentasGeneroRepetitivas
Nombre Programa: Programa2_VentasGeneroRepetitivas

```
1 // Programa para calcular el monto a pagar por un producto
2 // Autor: [Nombre]
3 // Fecha: [Fecha]
4 // Versión: 1.0
5
6 #include <iostream>
7 using namespace std;
8
9 // Función para calcular el monto a pagar
10 float Ventas(int cantidad, float precio)
11 {
12     return cantidad * precio;
13 }
14
15 // Función principal
16 int main()
17 {
18     int cantidad;
19     float precio;
20     float monto;
21
22     cout << "Ingrese cantidad: ";
23     cin >> cantidad;
24
25     cout << "Ingrese precio: ";
26     cin >> precio;
27
28     monto = Ventas(cantidad, precio);
29
30     cout << "Monto a pagar: " << monto << endl;
31
32     return 0;
33 }
```

Carpeta: Programas Semana 09
Nombre Proyecto: Proyecto4_VentasGeneroModulo
Nombre Programa: Programa4_VentasGeneroModulo

2. Elaborar un programa, que contenga los siguientes módulos Promedio(n1, n2, n3), Mayor(n1, n2, n3) y Menor(n1, n2, n3) en la que n1, n2 y n3 son números enteros; y que sean invocados en el módulo principal. Puedes usar función o procedimiento, según requiera el caso.
3. Elaborar un programa, que contenga los siguientes: módulos CalculaPromedioFinal(), el cual devuelve el valor calculado (ver fórmula) del promedio final de notas al módulo DeterminaCondicion() que muestra la condición según el promedio final de notas (ver tabla) y éste es invocado en el módulo principal. Puedes usar función o procedimiento, con o sin argumentos, según requiera el caso.

$$PF = C1 (20\%) + EP (25\%) + C2 (20\%) + EF (35\%)$$

Nota	Condición
De 15 a más	Con honores
De 10,5 a menos de 15	Aprobó
Menos de 10,5	Quedó

4. Elaborar un programa, para una empresa que otorga aumento al sueldo base de su trabajador (ver tabla). Se sugiere usar los módulos DeterminaPorcentaje(), el cual devuelve el valor del porcentaje según la tabla al módulo CalculaSueloTotal() que calcula y muestra el sueldo base, el monto de aumento y sueldo total; y éste es invocado en el módulo principal. Puedes usar función o procedimiento, con o sin argumentos, según requiera el caso.

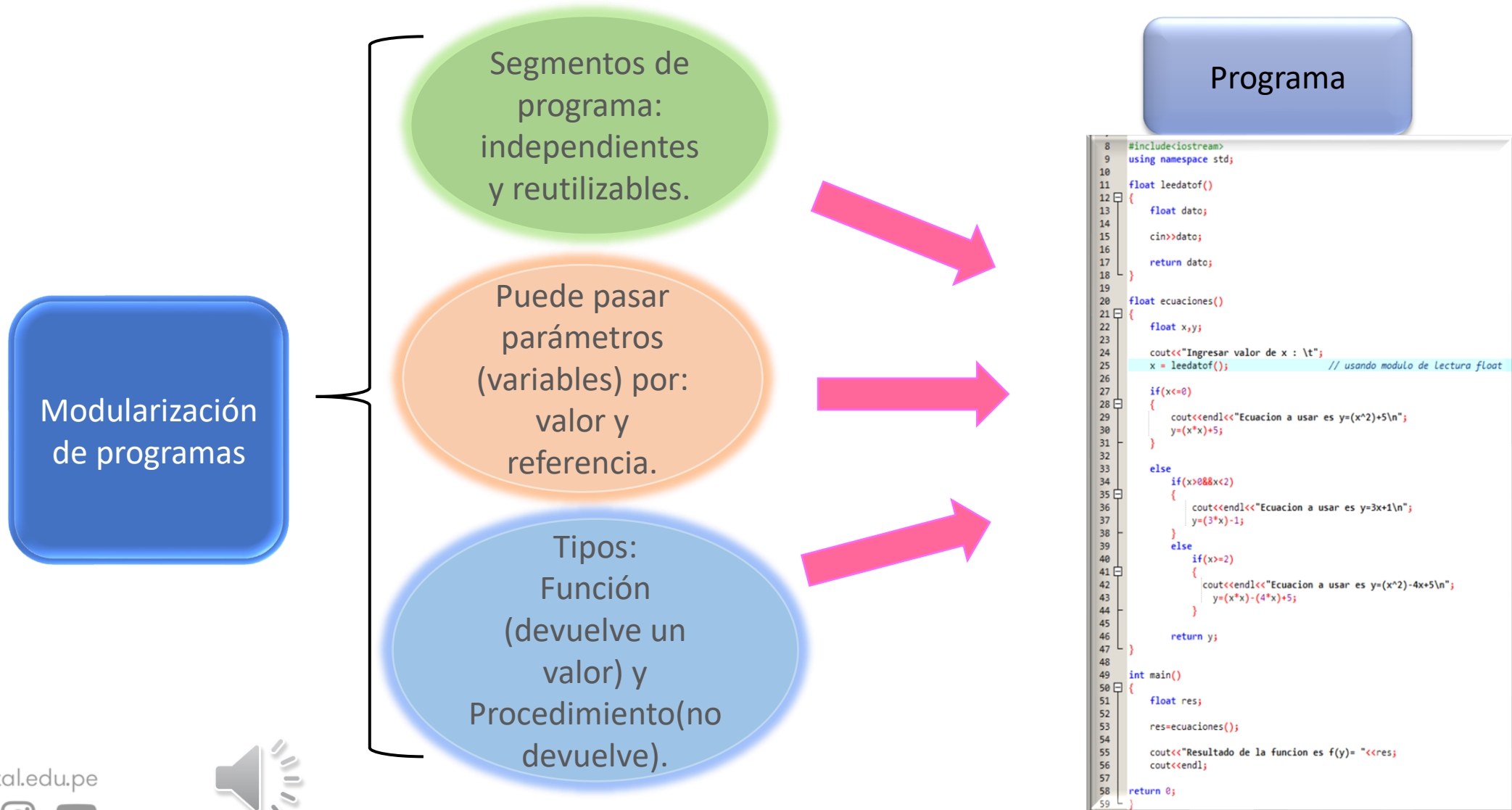
Sueldo (\$/.)	Aumento (%)
5001 - más	10
3501 - 5000	15
2001 - 3500	20
Hasta 2000	25

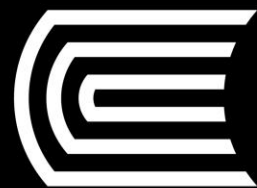
Carpeta: Programas Semana 09
Nombre Proyecto: Proyecto6_PromedioFinalModulo
Nombre Programa: Programa6_PromedioFinalModulo

Carpeta: Programas Semana 09
Nombre Proyecto: Proyecto7_AumentoSueldoModulo
Nombre Programa: Programa7_AumentoSueldoModulo



Conclusiones





ucontinental.edu.pe