



# Modularización de programas: tipos de módulos (función y procedimiento) y su aplicación en los programas

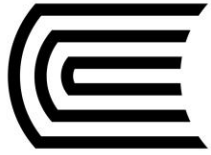
---



# Propósito

## Teoría

Reconoce los tipos de módulos (función y procedimiento) y su aplicación en los programas en el lenguaje de programación



# RECUERDA: Uso de Módulos

## Observa:

El módulo herencia tiene que calcular de uno hasta dos valores según el if-else (línea 28 y línea 34)

Por lo que se recomienda que sea procedimiento, y no función (ya que solo devuelve un valor)

```
1 #include <iostream>
2 using namespace std;
3
4 void herencia()
5 {
6     float her, may;
7     int cant;
8
9     do{
10         cout << "Ingresar importe de herencia \n";
11         cin >> her;
12
13         if (her <=0)
14             cout << "ERROR.La herencia debe ser mayor a 0";
15     } while (her <=0);
16
17     do{
18         cout << "Ingresar cantidad de hijos \n";
19         cin >> cant;
20
21         if (cant <= 0)
22             cout << "ERROR.La cantidad debe ser mayor a 0";
23     } while (cant <= 0);
24
25
26
27
28     if (cant < 4)
29     {
30         her = her / cant;
31         cout << "La herencia a cada hijo es" << her << endl;
32     }
33     else
34     {
35         may = her / 2;
36         her = may / (cant - 1);
37         cout << "\nLa herencia del hermano mayor es: " << may;
38         cout << "\nLa herencia de los demas hijos es: " << her;
39     }
40
41 }
42
43 int main()
44 {
45     herencia();
46
47     return 0;
48 }
```

ucontinental.edu.pe



```
1 #include <iostream>
2 using namespace std;
3
4 float leedatof()
5 {
6     float dato;
7
8     cin >> dato;
9
10    return dato;
11 }
12
13 int leedatof()
14 {
15     int dato;
16
17     cin >> dato;
18
19    return dato;
20 }
21
22 void herencia()
23 {
24     float her, may;
25     int cant;
26
27     do{
28         cout << "Ingresar importe de herencia:\t";
29         her = leedatof();
30
31         if (her <=0)
32             cout << "ERROR.La herencia debe ser mayor a 0";
33     } while (her <=0);
34
35     do{
36         cout << "Ingresar cantidad de hijos:\t";
37         cant = leedatof();
38
39         if (cant <= 0)
40             cout << "ERROR.La cantidad debe ser mayor a 0";
41     } while (cant <= 0);
42
43     if (cant < 4)
44     {
45         her = her / cant;
46         cout << "La herencia a cada hijo es:\t" << her << endl;
47     }
48     else
49     {
50         may = her / 2;
51         her = may / (cant - 1);
52         cout << "\nLa herencia del hermano mayor es: " << may;
53         cout << "\nLa herencia de los demas hijos es: " << her;
54     }
55
56 }
57
58 int main()
59 {
60     herencia();
61
62     return 0;
63 }
64 }
```

## Observa:

Se implementa un módulo para leer cada tipo de dato, y se invoca en otro módulo, por ejemplo: void herencia()



# RECUERDA:

## Uso de Variables Globales

```
1  #include <iostream>
2  using namespace std;
3
4  int creditos, contEstud, cont = 0, cantFem=0, cantMasc=0, acumCreditos=0;
5  int contFemPrimer=0, contFemSegund=0, contFemTercer=0, contFemCuart=0, contFemQuint=0;
6  int contMascPrimer=0, contMascSegund=0, contMascTercer=0, contMascCuart=0, contMascQuint=0;
7  int acumFemPrimer=0, acumFemSegund=0, acumFemTercer=0, acumFemCuart=0, acumFemQuint=0;
8  int acumMascPrimer=0, acumMascSegund=0, acumMascTercer=0, acumMascCuart=0, acumMascQuint=0;
9  char rpt, genero;
10 string mensaje;
11
12 int leedatoe()
13 {
14     int dato;
15     cin>>dato;
16     return dato;
17 }
18
19 char leedatoc()
20 {
21     char dato;
22     cin>>dato;
23     return dato;
24 }
25
26 void Mostrar()
27 {
28     cout<<"\nLa cantidad de estudiantes es:\t" << contEstud <<"\n";
29     cout<<"\nEl total acumulado de creditos es:\t" << acumCreditos <<"\n";
30     cout<<"\nLa cantidad de estudiantes femenino en primero es:\t" << contFemPrimer <<"\n";
31     cout<<"\nLa cantidad de estudiantes femenino en segundo es:\t" << contFemSegund <<"\n";
32     cout<<"\nLa cantidad de estudiantes femenino en tercero es:\t" << contFemTercer <<"\n";
33     cout<<"\nLa cantidad de estudiantes femenino en cuarto es:\t" << contFemCuart <<"\n";
34     cout<<"\nLa cantidad de estudiantes femenino en quinto es:\t" << contFemQuint <<"\n";
35     cout<<"\nLa cantidad de estudiantes masculino en primero es:\t" << contMascPrimer <<"\n";
36     cout<<"\nLa cantidad de estudiantes masculino en segundo es:\t" << contMascSegund <<"\n";
37     cout<<"\nLa cantidad de estudiantes masculino en tercero es:\t" << contMascTercer <<"\n";
38     cout<<"\nLa cantidad de estudiantes masculino en cuarto es:\t" << contMascCuart <<"\n";
39     cout<<"\nLa cantidad de estudiantes masculino en quinto es:\t" << contMascQuint <<"\n";
40     cout<<"\nEl acumulado de creditos femenino en primero es:\t" << acumFemPrimer <<"\n";
41     cout<<"\nEl acumulado de creditos femenino en segundo es:\t" << acumFemSegund <<"\n";
42     cout<<"\nEl acumulado de creditos femenino en tercero es:\t" << acumFemTercer <<"\n";
43     cout<<"\nEl acumulado de creditos femenino en cuarto es:\t" << acumFemCuart <<"\n";
44     cout<<"\nEl acumulado de creditos femenino en quinto es:\t" << acumFemQuint <<"\n";
45     cout<<"\nEl acumulado de creditos masculino en primero es:\t" << acumMascPrimer <<"\n";
46     cout<<"\nEl acumulado de creditos masculino en segundo es:\t" << acumMascSegund <<"\n";
47     cout<<"\nEl acumulado de creditos masculino en tercero es:\t" << acumMascTercer <<"\n";
48     cout<<"\nEl acumulado de creditos masculino en cuarto es:\t" << acumMascCuart <<"\n";
49     cout<<"\nEl acumulado de creditos masculino en quinto es:\t" << acumMascQuint <<"\n";
50 }
```

Variables Globales

# Ejemplo: Uso de Variables Globales



Antes  
semana 09

```
1 #include<iostream>
2 using namespace std;
3
4 int leedatoe()
5 {
6     int dato;
7     cin>>dato;
8     return dato;
9 }
10
11 void Promedio(int n1, int n2, int n3)
12 {
13     int sum;
14     double prom;
15     sum = n1 + n2 + n3;
16     prom = sum / 3;
17     cout<<"\nEl promedio es: "<< prom <<"\n\n";
18 }
19
20 void Mayor(int n1, int n2, int n3)
21 {
22     int may;
23     if(n1>=n2 && n1>=n3)
24         may = n1;
25     else if(n2>=n1 && n2>=n3)
26         may = n2;
27     else
28         may = n3;
29     cout<<"\nEl mayor es: "<< may <<"\n\n";
30 }
31
32 void Menor(int n1, int n2, int n3)
33 {
34     int men;
35     if(n1<=n2 && n1<=n3)
36         men = n1;
37     else if(n2<=n1 && n2<=n3)
38         men = n2;
39     else
40         men = n3;
41     cout<<"\nEl menor es: "<< men <<"\n\n";
42 }
43
44 int main()
45 {
46     int n1, n2, n3;
47     cout<<"Ingrese n1: ";
48     n1 = leedatoe();
49     cout<<"Ingrese n2: ";
50     n2 = leedatoe();
51     cout<<"Ingrese n3: ";
52     n3 = leedatoe();
53     Promedio(n1,n2,n3);
54     Mayor(n1,n2,n3);
55     Menor(n1,n2,n3);
56     return 0;
57 }
```

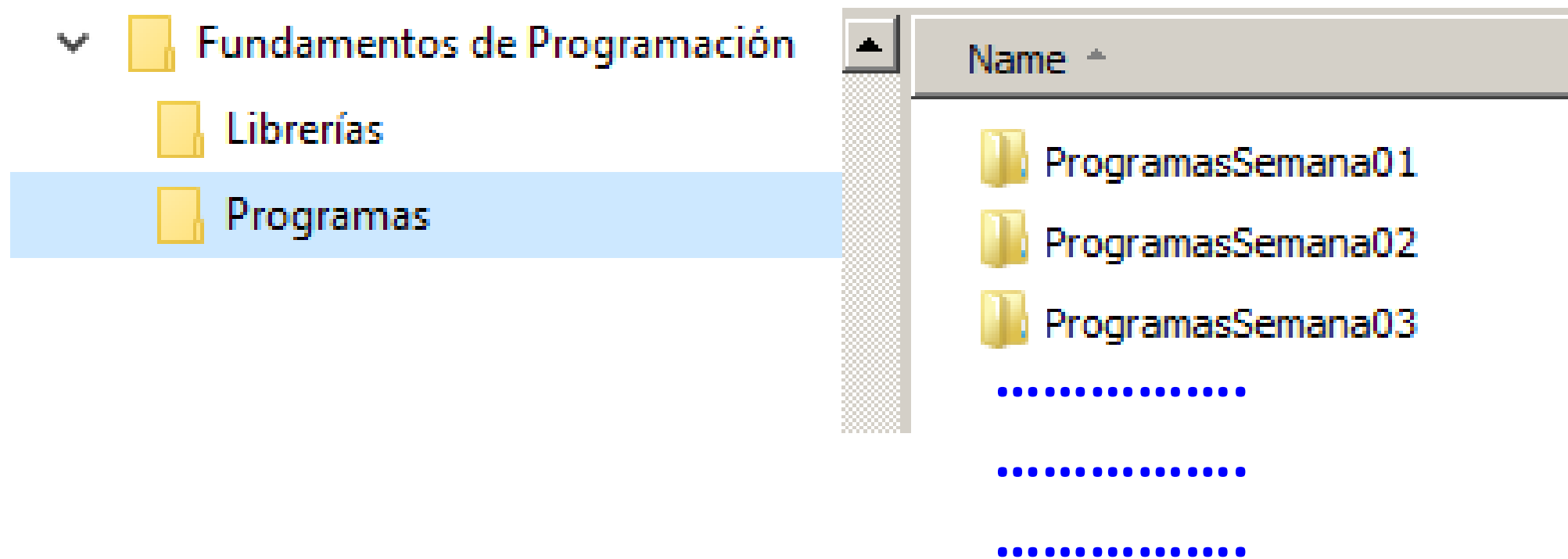
Declara  
Variables Locales

Ahora  
semana 10

```
1 #include<iostream>
2 using namespace std;
3
4 int n1, n2, n3;
5
6 int leedatoe()
7 {
8     int dato;
9     cin>>dato;
10     return dato;
11 }
12
13 void Promedio()
14 {
15     int sum;
16     double prom;
17     sum = n1 + n2 + n3;
18     prom = sum / 3;
19     cout<<"\nEl promedio es: "<< prom <<"\n\n";
20 }
21
22 void Mayor()
23 {
24     int may;
25     if(n1>=n2 && n1>=n3)
26         may = n1;
27     else if(n2>=n1 && n2>=n3)
28         may = n2;
29     else
30         may = n3;
31     cout<<"\nEl mayor es: "<< may <<"\n\n";
32 }
33
34 void Menor()
35 {
36     int men;
37     if(n1<=n2 && n1<=n3)
38         men = n1;
39     else if(n2<=n1 && n2<=n3)
40         men = n2;
41     else
42         men = n3;
43     cout<<"\nEl menor es: "<< men <<"\n\n";
44 }
45
46 int main()
47 {
48     cout<<"Ingrese n1: ";
49     n1 = leedatoe();
50     cout<<"Ingrese n2: ";
51     n2 = leedatoe();
52     cout<<"Ingrese n3: ";
53     n3 = leedatoe();
54     Promedio();
55     Mayor();
56     Menor();
57     return 0;
58 }
```

Declara  
Variables Globales

# Recomendación para crear los programas:



**Ahora, elaboremos los programas ejemplos..!**

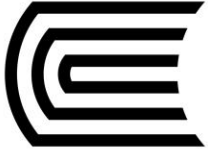


Carpeta: Programas Semana 10

Nombre Proyecto: Proyecto1\_HerenciaModulo

Nombre Programa: Programa1\_HerenciaModulo

```
1  #include <iostream>
2  using namespace std;
3
4  float leedatof()
5  {
6      float dato;
7
8      cin>>dato;
9
10     return dato;
11 }
12
13 int leedatoe()
14 {
15     int dato;
16
17     cin>>dato;
18
19     return dato;
20 }
21
22 void herencia()
23 {
24     float her, may;
25     int cant;
26
27     do{
28         cout << "Ingresar importe de herencia:\t";
29         her = leedatof();
30
31         if (her <= 0)
32             cout << "ERROR.La herencia debe ser mayor a 0";
33     } while (her <= 0);
34
35     do{
36         cout << "Ingresar cantidad de hijos:\t";
37         cant = leedatoe();
38
39         if (cant <= 0)
40             cout << "ERROR.La cantidad debe ser mayor a 0";
41
42     } while (cant <= 0);
43
44     if (cant < 4)
45     {
46         her = her / cant;
47         cout << "La herencia a cada hijo es:\t" << her << endl;
48     }
49     else
50     {
51         may = her / 2;
52         her = may / (cant - 1);
53         cout << "\nLa herencia del hermano mayor es: " << may;
54         cout << "\nLa herencia de los demas hijos es: " << her;
55     }
56 }
57
58
59 int main()
60 {
61     herencia();
62
63     return 0;
64 }
```



Carpeta: Programas Semana 10

Nombre Proyecto: Proyecto2\_IntercambioNumerosModulo

Nombre Programa: Programa2\_IntercambioNumerosModulo

Variables:

num1, num2 y temporal

Leer num1      num1 ← 5

5

num1

Leer num2      num2 ← 9

9

num2

Valores  
iniciales  
ingresados

temporal = num1      temporal ← 5

5

temporal

num1 = num2      num1 ← 9

9

num1

num2 = temporal      num2 ← 5

5

num2

Proceso de  
intercambio

Mostrar num1      num1 ← 9

9

num1

Mostrar num2      num2 ← 5

5

num2

Valores finales  
intercambiados

ucontinental.edu.pe



```
1 #include<iostream>
2 using namespace std;
3
4 int leedatoe()
5 {
6     int dato;
7     cin>>dato;
8     return dato;
9 }
10
11
12
13 void Intercambio()
14 {
15     int num1, num2, temp;
16
17     cout<<"Ingrese valor de num1: \t";
18     num1 = leedatoe();
19
20     cout<<"Ingrese valor de num2: \t";
21     num2 = leedatoe();
22
23     temp = num1;
24     num1 = num2;
25     num2 = temp;
26
27     cout<<"\n";
28     cout<<"El nuevo valor num1 es:\t" << num1 <<endl;
29     cout<<"El nuevo valor num2 es:\t" << num2 <<endl;
30 }
31
32
33 int main()
34 {
35     Intercambio();
36
37     return 0;
38 }
```





## Carpeta: Programas Semana 10

Nombre Proyecto: Proyecto3\_MayorMenorNumerosModulo

Nombre Programa: Programa3\_MayorMenorNumerosModulo

1. Leer el dato ingresado

cin>>datoIngresado;

18

20

5

may = 20

men = 5

2. Contar dato ingresado

cont = cont + 1;

cont = 0 + 1 = 1

cont = 1 + 1 = 2

cont = 2 + 1 = 3

3. Verificar si es el único dato ingresado

if (cont == 1)

if (1 == 1)

if (2 == 1)

if (3 == 1)

4. Si es Verdadero es el mayor y menor al mismo tiempo

```
{ may = datoIngresado;
  men = datoIngresado;
}
```

```
{ may = 18;
  men = 18;
}
```

```
{
}
```

```
{
}
```

5. Sino

else

else

else

5.1 Si dato ingresado es mayor que el mayor anterior

if (datoIngresado > may)

if (20 > 18)

if (5 > 20)

el mayor ahora será dato ingresado

may = datoIngresado

may = 20

5.2 Sino

else

else

5.2.1 Si dato ingresado es menor que el menor anterior

if (datoIngresado < men)

if (5 < 18)

el menor ahora será dato ingresado

men = datoIngresado

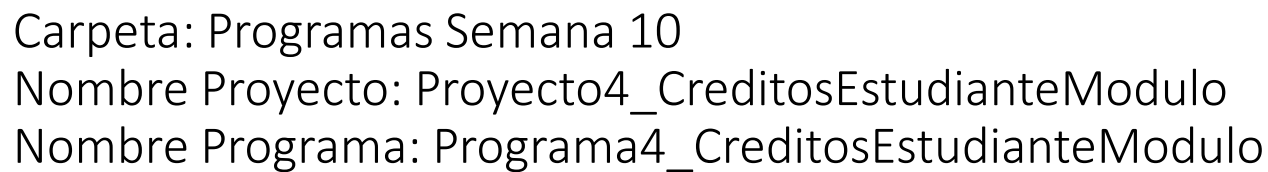
men = 5

REPETIR

```
1  #include <iostream>
2  using namespace std;
3
4  int cont = 0, may, men, datoIngresado;
5  char rpt;
6
7  int leedatoc()
8  {
9      int dato;
10     cin>>dato;
11     return dato;
12 }
13
14 char leedatoc()
15 {
16     char dato;
17     cin>>dato;
18     return dato;
19 }
20
21 void Mostrar()
22 {
23     cout<<"\nEl mayor es:\t" << may <<"\n";
24     cout<<"\nEl menor es:\t" << men <<"\n";
25 }
26
27 void MayorMenor()
28 {
29     do{
30         cout<<"\nIngrese valor:\t";
31         datoIngresado = leedatoc();
32         cont = cont + 1; // cont++;
33
34         if(cont == 1)
35         {
36             may = datoIngresado;
37             men = datoIngresado;
38         }
39         else
40         {
41             if(datoIngresado > may)
42                 may = datoIngresado;
43             else
44                 if(datoIngresado < men)
45                     men = datoIngresado;
46         }
47         do{
48             cout<<"\nDesea ingresar otros numero? (Ss - Nn):\t";
49             rpt = leedatoc();
50             if(rpt != 'S' && rpt != 's' && rpt != 'N' && rpt != 'n')
51                 cout<<"\nERROR. Vuelva a ingresar S o s o N o n.\n";
52             while(rpt != 'S' && rpt != 's' && rpt != 'N' && rpt != 'n') ;
53         }while(rpt == 'S' || rpt == 's');
54     }while(rpt == 'S' || rpt == 's');
55     Mostrar();
56 }
57
58 int main()
59 {
60     MayorMenor();
61     return 0;
62 }
```

Variables Globales

Se declara después de la cabecera, y se pueden utilizar en más de un módulo, manteniendo el valor y sin necesidad de volver a declararla.



Créditos acumulados	Año académico
Menos de 32	Primero
De 32 a 63	Segundo
De 64 a 95	Tercero
De 96 a 127	Cuarto
De 128 a más	Quinto



Variables  
Globales

```
52 void ProcesaCreditos()
```

```

rpta == 'S' || r
r();

```



# Aprendizaje Basado en Retos: ABR

(Con tu mismo grupo de trabajo)



**Idea General:**

**Los problemas de ingeniería que se presentan en tu entorno.**

Puede ser relacionados a  
tu carrera



**Pregunta esencial:**

**¿Qué programa puedo desarrollar para resolver un problema de ingeniería?**



**Reto:**

**Desarrollo de un programa orientado a un problema de ingeniería, en el lenguaje C++ aplicando la estructura modular del programa.**



# Aprendizaje Basado en Retos: ABR

(Con tu mismo grupo de trabajo)

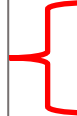
En el Aula Virtual Semana 10

## TAREA: Ideación

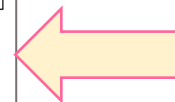
1. Descargar el enunciado



2. Responder a las preguntas guía



3. Enviar el archivo con las respuestas:  
aula virtual Semana 10 TAREA IDEACION,  
Grupo\_\_.docx.



Universidad Continental

Fundamentos de Programación  
Aprendizaje Basado en Retos (ABR)

Semana 10: Ideación  
TAREA 1 Aula Virtual

Tercera Unidad: Módulos para la programación: función y procedimiento

Sección : Docente :	Integrantes Grupo : Fecha : / / Duración: 100 min Tipo de actividad: Individual ( ) Grupal (X)
------------------------	--

Instrucciones: Lea detenidamente cada enunciado y desarrolle lo solicitado.

- Conformar grupo de 4 estudiantes.
- La presente actividad constituirá la calificación del examen de la semana 12.
- Analiza el RETO a desarrollar en las siguientes etapas:
  1. Semana 10: Ideación
  2. Semana 11: Prototipo-Implementación
  3. Semana 12: Publicación Producto Final-Reflexión

Consigna

I. Después de conformar el equipo de trabajo, lee detenidamente la idea general y la pregunta esencial.

IDEA GENERAL	Los problemas de ingeniería que se presentan en tu entorno.
PREGUNTA ESENCIAL	¿Qué programa puedo desarrollar para resolver un problema de ingeniería?

II. Lee y analiza el RETO propuesto:

RETO	Desarrollo de un programa orientado a un problema de ingeniería, en el lenguaje C++ aplicando la estructura modular del programa.
------	---

III. ¿Que necesitas para resolver el RETO?:

**PREGUNTAS GUÍA:**

1. ¿Qué estructuras de programación se usarían?
2. ¿Qué consideraciones debo tener para la validación del ingreso de datos?
3. ¿Qué tipo de módulos debe aplicarse para elaborar el programa?
4. ¿Qué librerías propias del lenguaje C++ pueden utilizarse?
5. ¿En qué casos debo usar menú de opciones para la elaboración del programa?

**ACTIVIDADES GUÍA:**

- Realización de actividades del Aula Virtual (videos, ppts, foros) de la unidad
- Desarrollo de la Guía Práctica
- Revisión de otros ejercicios resueltos.

**RECURSOS GUÍA, que te ayudarán a responder las preguntas guía:**

VIDEO

18- Programación en C – Modularización - Facu Darfe  
<https://youtu.be/va8g5IEsOKg>

LIBROS:

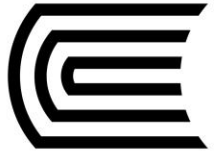
[https://estudiantesavp.ucontinental.edu.pe/pluginfile.php/2300152/mod\\_resource/content/0/joyanes\\_pp202-203.pdf](https://estudiantesavp.ucontinental.edu.pe/pluginfile.php/2300152/mod_resource/content/0/joyanes_pp202-203.pdf)  
[https://estudiantesavp.ucontinental.edu.pe/pluginfile.php/2300153/mod\\_resource/content/0/joyanes\\_pp210-212.pdf](https://estudiantesavp.ucontinental.edu.pe/pluginfile.php/2300153/mod_resource/content/0/joyanes_pp210-212.pdf)

IDDE Dev C

<https://sourceforge.net/projects/orwelldevcpp/>

ucontinental.edu.pe





# Conclusiones

Modularización de programas: tipos de módulos y su aplicación en los programas

Algunos módulos deben ser función, como los lee datos.

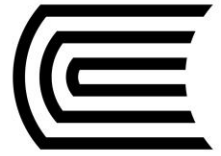
El módulo es procedimiento si debe calcular y mostrar mas de un valor.

El módulo principal invoca a un módulo que realiza el algoritmo de solución.

Programa

```
1 #include <iostream>
2 using namespace std;
3
4 float leedatof()
5 {
6     float dato;
7     cin>>dato;
8     return dato;
9 }
10
11 int leedatoe()
12 {
13     int dato;
14     cin>>dato;
15     return dato;
16 }
17
18 void herencia()
19 {
20     float her, may;
21     int cant;
22
23     do{
24         cout << "Ingresar importe de herencia:\t";
25         her = leedatof();
26         if (her <=0)
27             cout << "ERROR.La herencia debe ser mayor a 0";
28     } while (her <=0);
29
30     do{
31         cout << "Ingresar cantidad de hijos:\t";
32         cant = leedatof();
33         if (cant <= 0)
34             cout << "ERROR.La cantidad debe ser mayor a 0";
35     } while (cant <= 0);
36
37     if (cant < 4)
38     {
39         her = her / cant;
40         cout << "La herencia a cada hijo es:\t" << her << endl;
41     }
42     else
43     {
44         may = her / 2;
45         her = may / (cant - 1);
46         cout << "\nLa herencia del hermano mayor es: " << may;
47         cout << "\nLa herencia de los demas hijos es: " << her;
48     }
49 }
50
51 int main()
52 {
53     herencia();
54     return 0;
55 }
```





# Propósito

## Práctica

Crea programas con tipos de módulos (función y procedimiento), en el lenguaje de programación C/ C++.



# Desarrollamos la Guía Práctica 10

Guía práctica N°10

## 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, para una empresa que otorga aumento al sueldo base de su trabajador (ver tabla). Además de los módulos que leen datos, y los módulos que procesan (que calculan o determina valores), crear un módulo ReporteFinal(), para mostrar la cantidad total de trabajadores, la cantidad de trabajadores que recibieron aumento de 10%, la cantidad de trabajadores que recibieron aumento de 15%, la cantidad de trabajadores que recibieron aumento de 20%, y la cantidad de trabajadores que recibieron aumento de 25%, además del acumulado de sueldos totales. Usar variables globales, función o procedimiento, con o sin argumentos, según requiera el caso. Referencia de solución:

Carpeta: Programas Semana 09  
Nombre Proyecto: Proyecto7\_AumentoSueldoModulo  
Nombre Programa: Programa7\_AumentoSueldoModulo

Sueldo (\$/.)	Aumento (%)
\$001 - más	10
\$501 - \$1000	15
\$1001 - \$1500	20
Más de \$1500	25

2. Elaborar un programa, para ingresar género y estado civil de una persona, así como su respectiva nota (0 a 20) y, además de los módulos que leen datos, y los módulos que procesan (que calculan o determina valores), crear el módulo ReporteFinalEstudiante() para mostrar la siguiente información. Usar variables globales, función o procedimiento, con o sin argumentos, según requiera el caso.

Mostrar:

La cantidad de hombres aprobados.  
La cantidad de mujeres aprobadas.  
La cantidad de hombres casados desaprobados.  
La cantidad de mujeres solteras aprobadas.  
La sumatoria de todas las notas ingresadas.  
El promedio de notas ingresadas.

Referencia de Solución:

Carpeta: Programas Semana 07  
Nombre Proyecto: Proyecto4\_NotasContaAcumulaMas  
Nombre Programa: Programa4\_NotasContaAcumulaMas

3. Elaborar un programa para gestionar las facturas, para una empresa que se dedica a la venta de desinfectantes. Para cada factura se ingresa: el código del artículo, la cantidad vendida en litros y el precio por litro. Además de los módulos que leen datos, y los módulos que procesan (que calculan o determina valores), crear un módulo para mostrar por cada factura: el monto total a facturar (monto de pago), y además crear un módulo para mostrar la cantidad total de facturas ingresadas, la cantidad de facturas con monto total a facturar de más de 600, el acumulado de monto total a facturar, y el acumulado de litros vendidos. Usar variables globales, función o procedimiento, con o sin argumentos, según requiera el caso

4. Elaborar un programa que determine el IMC (Índice de masa corporal) de una persona al ingresar su peso en kg y su talla en metros ( $IMC = \text{Peso} / \text{Talla}^2$ ), y por cada persona mostrar su estado según la siguiente tabla:

Resultado IMC	Estado
Menos de 18.50	Infra Peso
De 18.50 a 24.99	Peso Normal
De 25 a 29.99	Sobre Peso
De 30 a 34.99	Obesidad leve
De 35 a 39.99	Obesidad media
De 40 a más	Obesidad mórbida

Para la validación de datos, se considera:  
Peso mínimo: 38 kg - Peso máximo: 260 kg.  
Talla mínima 0.90 m - Talla máxima 2.20 m.

Además, crear un módulo para mostrar la cantidad de personas ingresadas, la cantidad de personas con estado "Peso Normal", la cantidad de personas con "Obesidad Media" y la cantidad de personas con "Obesidad Mórbida".

Usar variables globales, función o procedimiento, con o sin argumentos, según requiera el caso.

Carpeta: Programas Semana 10

Nombre Proyecto: Proyecto5\_AumentoSueldoModuloV2

Nombre Programa: Programa5\_AumentoSueldoModuloV2

Carpeta: Programas Semana 10

Nombre Proyecto: Proyecto6\_ContaAcumulaModulo

Nombre Programa: Programa6\_ContaAcumulaModulo

Carpeta: Programas Semana 10

Nombre Proyecto: Proyecto7\_DesinfectantesModulo

Nombre Programa: Programa7\_DesinfectantesModulo

Carpeta: Programas Semana 10

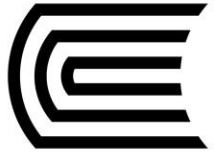
Nombre Proyecto: Proyecto8\_MasaCorporalModulo

Nombre Programa: Programa8\_MasaCorporalModulo

ucontinental.edu.pe







# Conclusiones

Modularización de programas: tipos de módulos y su aplicación en los programas

Algunos módulos deben ser función, como los lee datos.

El módulo es procedimiento si debe calcular y mostrar mas de un valor.

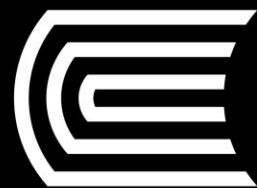
El módulo principal invoca a un módulo que realiza el algoritmo de solución.

Programa

```
1 #include <iostream>
2 using namespace std;
3
4 float leedatof()
5 {
6     float dato;
7     cin>>dato;
8     return dato;
9 }
10
11 int leedatof()
12 {
13     int dato;
14     cin>>dato;
15     return dato;
16 }
17
18 void herencia()
19 {
20     float her, may;
21     int cant;
22
23     do{
24         cout << "Ingresar importe de herencia:\t";
25         her = leedatof();
26         if (her <=0)
27             cout << "ERROR.La herencia debe ser mayor a 0";
28     } while (her <=0);
29
30     do{
31         cout << "Ingresar cantidad de hijos:\t";
32         cant = leedatof();
33         if (cant <= 0)
34             cout << "ERROR.La cantidad debe ser mayor a 0";
35     } while (cant <= 0);
36
37     if (cant < 4)
38     {
39         her = her / cant;
40         cout << "La herencia a cada hijo es:\t" << her << endl;
41     }
42     else
43     {
44         may = her / 2;
45         her = may / (cant - 1);
46         cout << "\nLa herencia del hermano mayor es: " << may;
47         cout << "\nLa herencia de los demas hijos es: " << her;
48     }
49 }
50
51 int main()
52 {
53     herencia();
54     return 0;
55 }
```







**ucontinental.edu.pe**