



Modularización de programas: Funciones propias del lenguaje de programación: definición



Propósito

Teoría

Reconoce la sintaxis de las funciones propias del lenguaje de programación



RECUERDA: Funciones Propias

```
1 #include<iostream>
2 #include<ctype.h>
3 #include<math.h>
4 using namespace std;
5
6 char categ;
7 int num1, num2, z;
8
9 void Cacular()
10 {
11     cout<<"Ingrese categoria (A a B b C c): ";
12     cin>>categ;
13
14     categ = toupper(categ);
15
16     cout<<"Ingrese num1: ";
17     cin>>num1;
18
19     cout<<"Ingrese num2: ";
20     cin>>num2;
21
22     num1 = abs(num1);
23     num2 = abs(num2);
24
25     switch(categ)
26     {
27         case 'A':{ z = pow(num1,num2); }break;
28
29         case 'B':{ z = sqrt(num2) + pow(num2,num1); }break;
30
31         case 'C':{ z = sqrt(num1+num2); }break;
32     }
33
34     cout<<"El resultado es: " << z <<endl;
35
36 }
37
38 int main()
39 {
40     Cacular();
41
42     return 0;
43 }
```

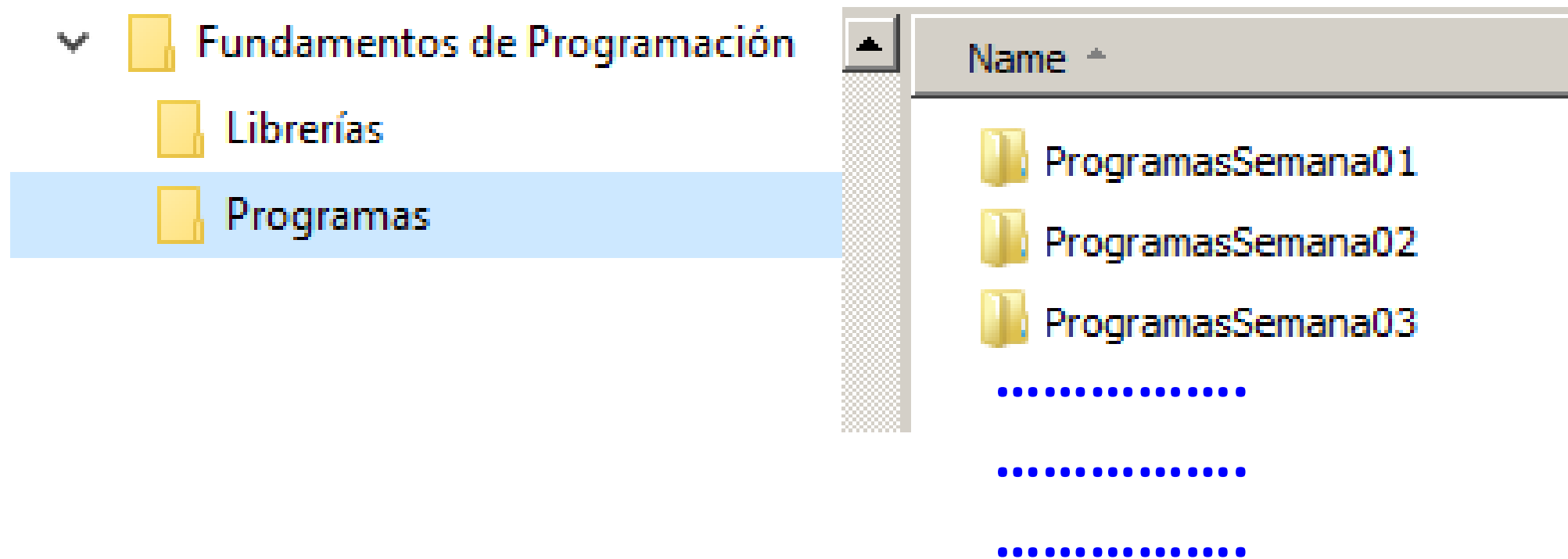
Uso de las librerías:

math.h
ctype.h

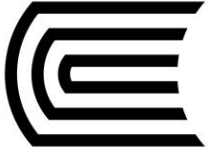
Invoca a la función **toupper**
(convierte minúsculas a mayúsculas)

Invoca a la función **abs**
(devuelve el valor absoluto de un número)

Recomendación para crear los programas:



Ahora, elaboremos los programas ejemplos..!



Carpeta: Programas Semana 13

Nombre Proyecto: Proyecto1_DiferenciaLetrasNumerosFunciones

Nombre Programa: Programa1_DiferenciaLetrasNumerosFunciones

```
1 #include <iostream>
2 #include "LecturaDatos.h"
3 #include <stdlib.h> // Librería que invoca a atof() o atoi()
4 #include <string>
5 using namespace std;
6
7 char repeticion()
8 {
9
10     char letra, rpt;
11
12     do
13     {
14         cout << "Desea continuar? (S/N):\t\t\t";
15         letra = leedatos();
16         if (isalpha(letra))
17         {
18             rpt = toupper(letra);
19             if (rpt != 'S' && rpt != 'N')
20                 cout << "Error! Ingrese solo S o N. Vuelva a Ingresar." << "\n\n";
21         }
22         else
23         {
24             cout << "Solo debe ingresar letras." << "\n\n";
25             rpt = 'T';
26         }
27     } while (rpt != 'S' && rpt != 'N');
28
29     cout << "\n";
30     return rpt;
31 }
32
33
34
35 double ventas()
36 {
37     int cantart;
38     double precio, dcto, MontBruto, montdsct, montofinal;
39     string CadenaIngresada; // usando tipo de dato string
40
41     do{
42         cout << "\n Ingresar el precio de articulo: \t";
43         CadenaIngresada=leedatos();
44         if (isalpha(CadenaIngresada[0]))
45         {
46             precio = -5; // valor cualquiera
47             cout << "\t\t ERROR. Es letra.\n";
48         }
49         else
50         {
51             precio = atof(CadenaIngresada.c_str());
52             // usando string...permite convertir con la funcion c_str()
53
54             if (precio <= 0)
55                 cout << "\t\t ERROR. Tiene que ser un numero positivo.\n";
56
57         }
58     } while (precio <= 0);
59
60 }
```

```
61
62 do{
63     cout << "\n Ingresar la cantidad de articulos: \t";
64     CadenaIngresada=leedatos();
65     if (isalpha(CadenaIngresada[0]))
66     {
67         cantart = -5; // valor cualquiera
68         cout << "\t\t ERROR. Es letra.\n";
69     }
70     else
71     {
72         cantart = atoi(CadenaIngresada.c_str());
73         // usando string...permite convertir con la funcion c_str()
74
75         if (cantart <= 0)
76             cout << "\t\t ERROR. Tiene que ser un numero positivo.\n";
77     }
78 } while (cantart <= 0);
79
80
81 if (cantart > 100)
82 {
83     dcto = 0.4;
84 }
85 else
86 {
87     if (cantart >= 25 && cantart <= 100)
88     {
89         dcto = 0.2;
90     }
91     else
92     {
93         if (cantart >= 10 && cantart <= 25)
94         {
95             dcto = 0.1;
96         }
97         else
98         {
99             if (cantart < 10)
100             {
101                 dcto = 0.0;
102                 cout << "\n\n";
103                 cout << "\t\t No tiene descuento.\n";
104                 cout << "\n\n";
105             }
106
107             MontBruto = precio * cantart;
108             montdsct = MontBruto * dcto;
109             montofinal = MontBruto - montdsct;
110
111             cout << "\n\n";
112             cout << "Monto sin descuento es:\t\t\t" << MontBruto;
113             cout << "\n\n";
114             cout << "Monto de descuento es:\t\t\t" << montdsct;
115             cout << "\n\n";
116             cout << "Monto a pagar es:\t\t\t" << montofinal;
117             cout << "\n\n";
118
119             return montofinal;
120 }
```

```
122 int main()
123 {
124     char rpt;
125     int cont = 0;
126     double Totalvendido = 0;
127     double montofinal;
128
129     do{
130         montofinal = ventas();
131         cont++; // contador de ventas
132         Totalvendido = Totalvendido + montofinal; // acumulador de ventas
133         rpt = repeticion();
134     } while (rpt == 'S');
135
136     cout << "\n\n";
137
138     cout << "Acumulador de Venta es:\t\t\t" << Totalvendido;
139     cout << "\n";
140
141     cout << "Contador de Ventas es:\t\t\t" << cont;
142     cout << "\n";
143     cout << "\n\n";
144
145     system("pause");
146     return 0;
147 }
148
149
150 }
```



Carpeta: Programas Semana 12

Nombre Proyecto: Proyecto2_CajeroModuloFunciones

Nombre Programa: Programa2_CajeroModuloFunciones

```
1 #include<iostream>
2 #include "LecturaDatos.h"
3 using namespace std;
4
5 double saldo = 0, depos, retir;
6 int rpt, op, opc;
7
8 void Depositar()
9 {
10     do{
11         cout << "\n\t\tIngrese monto a depositar:\t";
12         depos = leedatod();
13         if (depos <= 0)
14             cout << "\t\t ERROR.Vuelva a ingresar " << endl;
15     }while (depos <= 0);
16
17     saldo = saldo + depos;
18 }
19
20 void Retirar()
21 {
22     do{
23         cout << "\n\t\tIngrese monto a retirar:\t";
24         retir = leedatod();
25         if (retir <= 0)
26             cout << "\t\t ERROR.Vuelva a ingresar " << endl;
27     }while (retir <= 0);
28
29     if (retir <= saldo)
30         saldo = saldo - retir;
31     else
32         cout << "\t\tNo se puede. No tiene saldo."<< endl;
33 }
34
35 void VerSaldo()
36 {
37     cout << endl;
38     cout << "\t\tSu saldo es:\t\t\t" << saldo << endl;
39 }
40
41 void Salida()
42 {
43     cout << "\n\t\t GRACIAS. Vuelva pronto..!!\n";
44 }
```

Usar isalpha() y atof()
para validar los datos
ingresados.

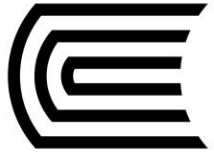
```
46 void Procesar() // Sub Menú
47 {
48     cout << endl << endl;
49     cout << "\t\t\t-----CAJERO-----\t\t" << endl;
50     cout << endl;
51     cout << "\t\t1.Depositar." << endl;
52     cout << "\t\t2.Retirar." << endl;
53     cout << "\t\t3.Saldo." << endl;
54     cout << "\t\t4.Salir." << endl;
55     cout << endl;
56
57     cout << "\t\t Ingrese opcion:\t";
58     opc = leedatod();
59     if (opc < 1 || opc > 4)
60         cout << "\t\t ERROR.Vuelva a Ingresar";
61     while (opc < 1 || opc > 4);
62
63     switch (opc)
64     {
65         case 1: { Depositar();
66                 VerSaldo();
67                 cout << endl;
68                 }break;
69
70         case 2: { Retirar();
71                 VerSaldo();
72                 cout << endl;
73                 }break;
74
75         case 3: { VerSaldo();
76                 cout << endl;
77                 }break;
78
79         case 4: { Salida();
80                 }break;
81     }
82 }
83
84 int Menu() // Menú principal
85 {
86     cout << endl << endl;
87     cout << "\t\t\t-----MENU-----\t\t" << endl;
88     cout << endl;
89     cout << "\t\t1.Procesar." << endl;
90     cout << "\t\t2.Salir." << endl;
91     cout << endl;
92     cout << "\t\t Ingrese opcion:\t";
93     op = leedatod();
94     if (op != 1 && op != 2)
95         cout << "\t\t ERROR.Vuelva a Ingresar";
96     while (op != 1 && op != 2);
97
98     switch (op)
99     {
100         case 1: { do{
101                   Procesar();
102                   cout << endl;
103                   }while (opc != 4);
104                 }break;
105
106         case 2: { Salida();
107                 }break;
108     }
109     return op;
110 }
```

```
112 int main()
113 {
114     do{
115         rpt = Menu();
116     }while (rpt != 2);
117
118     return 0;
119 }
120 }
```



en Word	Que hace?	Programa en C/C++ ejemplificando su uso (máximo 12 líneas de código y de autoría propia)	Prueba de escritorio ppts animadas aspa
			X
			X
			X
			X
			X
			X
			X
			X
			X
			X
			X
			X
			X
			X
			X
			X

	Trabajador Calificado		
Puntaje Evaluación	Profesional (P)	Técnico (T)	Obrero (O)
Menor a 10	45%	25%	5%
10 – 30	25%	15%	3%
31 - 70	20%	10%	2%
71- 100	"Es Observado"	"Es Renovado"	"Es Promovido"



Conclusiones

Modularización
de programas:
Funciones
propias del
lenguaje de
programación

Funciones
proporcionadas
por el propio
lenguaje de
programación.

Estas funciones
simplifican
operaciones.

Cada función
depende una
librería
proporcionadas
por el lenguaje de
programación.

Programa

```
1 #include <iostream>
2 #include "LecturaDatos.h"
3 #include <stdlib.h> // librería que invoca a atof() o atoi()
4 #include <string>
5 using namespace std;
6
7 char repeticion()
8 {
9     char letra, rptas;
10
11     do
12     {
13         cout << "Desea continuar? (S/N):\t\t\t";
14         letra = leedatoc();
15
16         if (isalpha(letra))
17         {
18             rptas = toupper(letra);
19             if (rptas != 'S' && rptas != 'N')
20                 cout << "Error! Ingrese solo S o N. Vuelva a Ingresar." << "\n\n";
21         }
22         else
23         {
24             cout << "Solo debe ingresar letras." << "\n\n";
25             rptas = 'T';
26         }
27     } while (rptas != 'S' && rptas != 'N');
28
29     cout << "\n";
30     return rptas;
31 }
32
33
34 double ventas()
35 {
36     int cantart;
37     double precio, dcto, MontBruto, montdct, montofinal;
38     string CadenaIngresada; // usando tipo de dato string
39
40     do{
41         cout << "\n Ingresar el precio de articulo: \t";
42         CadenaIngresada=leedatos();
43
44         if (isalpha(CadenaIngresada[0]))
45         {
46             precio = -5; // valor cualquiera
47             cout << "\t\t\t ERROR. Es letra.\n";
48         }
49         else
50         {
51             precio = atof(CadenaIngresada.c_str());
52             // usando string...permite convertir con la funcion c_str()
53             if (precio <= 0)
54                 cout << "\t\t\t ERROR. Tiene que ser un numero positivo.\n";
55         }
56     } while (precio <= 0);
57
58 }
```





Propósito

Práctica

Crea programas con funciones propias del lenguaje, más usadas, en el lenguaje de programación C/ C++.



Desarrollamos la Guía Práctica 13

Guía práctica N°13

Fundamentos de Programación

Cuarta Unidad: Módulos para la programación:

funciones propias del lenguaje de programación – recursividad

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 permita ingresar el valor del promedio final de una nota, validando que esté entre cero a veinte y mostrar para la cantidad de promedios finales ingresados, el acumulado de promedios finales, y el promedio general de los promedios finales. Puedes usar funciones isalpha, atoi, librerías de programación para leer datos, variables globales, menú principal, sub menú, función o procedimiento, con o sin argumentos, según requiera el caso.
2. Elaborar un programa, para bonificar a sus empleados, según la situación laboral de su esposa, y la cantidad de hijos, y los hijos que están en universidad. Mostrar la cantidad de esposas desempleadas y cuantos trabajadores no tienen esposa y cuantos no tienen hijos. Puedes usar funciones isalpha, atoi, librerías de programación para leer datos, variables globales, menú principal, sub menú, función o procedimiento, con o sin argumentos, según requiera el caso.

Carpeta: Programas Semana 13

Nombre Proyecto: Proyecto3_PromediosModuloFunciones

Nombre Programa: Programa3_PromedioModuloFunciones

Carpeta: Programas Semana 13

Nombre Proyecto: Proyecto4_EsposaHijosModuloFunciones

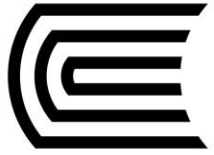
Nombre Programa: Programa4_EsposaHijosModuloFunciones

ucontinental.edu.pe



Referencias bibliográficas consultadas y/o enlaces recomendados

- JOYANES, L. (2008). Fundamentos de Programación. Algoritmos, estructura de datos y objetos, 4ª Edición. Madrid: McGraw-Hill.
- Beekman, G., Pacheco, R. y Tábor, A. (2008). Introducción a la computación. México: Pearson Educación.



Conclusiones

Modularización
de programas:
Funciones
propias del
lenguaje de
programación

Funciones
proporcionadas
por el propio
lenguaje de
programación.

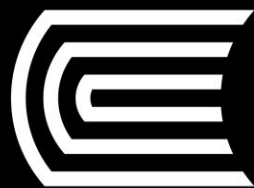
Estas funciones
simplifican
operaciones.

Cada función
depende una
librería
proporcionadas
por el lenguaje de
programación.

Programa

```
1 #include <iostream>
2 #include "LecturaDatos.h"
3 #include <stdlib.h> // librería que invoca a atof() o atoi()
4 #include <string>
5 using namespace std;
6
7 char repeticion()
8 {
9     char letra, rptas;
10
11     do
12     {
13         cout << "Desea continuar? (S/N):\t\t\t";
14         letra = leedatoc();
15
16         if (isalpha(letra))
17         {
18             rptas = toupper(letra);
19             if (rptas != 'S' && rptas != 'N')
20                 cout << "Error! Ingrese solo S o N. Vuelva a Ingresar." << "\n\n";
21         }
22         else
23         {
24             cout << "Solo debe ingresar letras." << "\n\n";
25             rptas = 'T';
26         }
27     } while (rptas != 'S' && rptas != 'N');
28
29     cout << "\n";
30     return rptas;
31 }
32
33
34 double ventas()
35 {
36     int cantart;
37     double precio, dcto, MontBruto, montdct, montofinal;
38     string CadenaIngresada; // usando tipo de dato string
39
40     do{
41         cout << "\n Ingresar el precio de articulo: \t";
42         CadenaIngresada=leedatos();
43
44         if (isalpha(CadenaIngresada[0]))
45         {
46             precio = -5; // valor cualquiera
47             cout << "\t\t\t ERROR. Es letra.\n";
48         }
49         else
50         {
51             precio = atof(CadenaIngresada.c_str());
52             // usando string...permite convertir con la funcion c_str()
53             if (precio <= 0)
54                 cout << "\t\t\t ERROR. Tiene que ser un numero positivo.\n";
55         }
56     } while (precio <= 0);
57
58 }
```





ucontinental.edu.pe