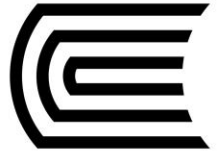




Modularización de programas: Funciones propias del lenguaje de programación y recursividad, con librerías de programación



Propósito

Teoría

Reconoce la sintaxis de las librerías de programación con Funciones propias del lenguaje de programación y recursividad, en el lenguaje de programación



RECUERDA: Librerías

LecturaDatos.h

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4
5
6  int leedatoe()
7  {
8      int dato;
9
10     cin>> dato;
11
12     return dato;
13 }
14
15 float leedatof()
16 {
17     float dato;
18
19     cin>> dato;
20
21     return dato;
22 }
23
24 double leedatod()
25 {
26     double dato;
27
28     cin>> dato;
29
30     return dato;
31 }
32
33 char leedatoc()
34 {
35     char dato;
36
37     cin>> dato;
38
39     return dato;
40 }
41
42 string leedatos()
43 {
44     string dato;
45
46     cin>> dato;
47
48     return dato;
49 }
```


FRecursivas.h

```
1  #include<iostream>
2  using namespace std;
3
4  int factorial(int n)
5  {
6      if(n==0)
7          return 1;
8      else
9          return n*factorial(n - 1);
10 }
11
12 int potencia(int b, int e)
13 {
14     if(e==0)
15         return 1;
16     else
17         return b* potencia(b, e - 1);
18 }
19
20
21 int multiplica(int a, int b)
22 {
23     if(a == 0 || b == 0)
24         return 0;
25     else
26         if(b == 1)
27             return a;
28         else
29             return a + multiplica(a, b - 1);
30 }
31
32
33 int fibonaci(int n)
34 {
35     if(n==1 || n==2)
36     {
37         return 1;
38     }
39     else
40     {
41         return fibonaci(n-1)+fibonaci(n-2);
42     }
43 }
44
45 int division(int a, int b)
46 {
47     if(b>a)
48         return 0;
49     else
50         return division(a-b,b)+1;
51 }
52 }
```



Recomendación crear librería FRecurativas:

▼  Fundamentos de Programación

 Librerías

 Programas

Ahora, elaboremos los programas ejemplos..!

ucontinental.edu.pe



ConvierteDatos.h

```
1 #include <iostream>
2 #include <stdlib.h>
3 #include <string>
4 using namespace std;
5
6 float leedatofPositivos()
7 {
8     float dato;
9     string cadIngreso;
10
11     cin >> cadIngreso;
12
13     if (isalpha(cadIngreso[0]))
14     {
15         dato = -5;
16         cout << "Error. Es letra.\n";
17     }
18     else
19     {
20         dato = atof(cadIngreso.c_str());
21         if(dato < 0)
22             cout << "Error, debe ser un numero positivo.\n";
23     }
24     return dato;
25 }
26
27 int leedatoePositivos()
28 {
29     int dato;
30     string cadIngreso;
31
32     cin >> cadIngreso;
33
34     if (isalpha(cadIngreso[0]))
35     {
36         dato = -5;
37         cout << "Error. Es letra.\n";
38     }
39     else
40     {
41         dato = atoi(cadIngreso.c_str());
42         if(dato < 0)
43             cout << "Error, debe ser un numero positivo.\n";
44     }
45     return dato;
46 }
47
48 }
```



Carpeta: Programas Semana 15

Nombre Proyecto: Proyecto1_EsposaHijosLibreriasRecursion

Nombre Programa: Programa1_EsposaHijosLibreriasRecursion

```
1 #include<iostream>
2 #include<string>
3 #include<ctype.h>
4 #include<windows.h>
5 #include "ConvierteDatos.h"
6 #include "LecturaDatos.h"
7 using namespace std;
8
9 void proceso()
10 {
11     int opc_esposa,opc_hijos,opc_universidad;
12     int estado_esposa,bono_esposa=0,cant_hijos,bono_hijo=0,bono_universidad=0,hijos_universidad;
13     float sueldo=0,sueldo_basico=0;
14     char genero;
15     string nombreTrab;
16
17     cout<<"\tIngrese Nombre Trabajador:\t";
18     nombreTrab=leedatos();
19
20     do{
21         cout<<"\tIngrese Genero (F o M):\t";
22         genero=leedatoc();
23         if (isalpha(genero))
24         {
25             genero = toupper(genero);
26             if (genero != 'F' && genero != 'M')
27             {
28                 cout << "\tError! Ingrese solo F o M. Vuelva a Ingresar." << "\n\n";
29             }
30             else
31             {
32                 cout << "\tSolo debe ingresar letras." << "\n\n";
33                 genero = 'T';
34             }
35             while (genero != 'F' && genero != 'M');
36         }
37         do{
38             cout<<"\tIngrese su sueldo: \t";
39             sueldo=leedatofPositivos();
40             if(sueldo<0)
41             {
42                 cout<<"\tERROR. vuelva a ingresar, el sueldo debe ser mayor a 0."<<"\n\n";
43             }
44         }while(sueldo<0);
45
46         do{
47             cout<<"\n\n\tTiene Esposa?\n"<<endl;
48             cout<<"\t1. SI"<<endl;
49             cout<<"\t2. NO"<<endl;
50             cout<<"\n";
51             cout<<"\tIngrese opcion(1 o 2): \t";
52             opc_esposa=leedatofPositivos();
53             if(opc_esposa<1 || opc_esposa>2)
54             {
55                 cout<<"\tERROR. vuelva a ingresar, la opcion es 1 o 2";
56             }while(opc_esposa<1 || opc_esposa>2);
57
58             switch (opc_esposa)
59             {
60                 case 1 :{ do{
61                     cout<<"\n\n\tEstado de Esposa"<<endl;
62                     cout<<"\n";
63                     cout<<"\t1. Empleada"<<endl;
64                     cout<<"\t2. Desempleada"<<endl;
65                     cout<<"\n";
66                     cout<<"\tIngrese opcion(1 o 2): \t";
67                     estado_esposa=leedatofPositivos();
68                     if(estado_esposa<1 || estado_esposa>2)
69                     {
70                         cout<<"\tERROR. vuelva a ingresar, la opcion es 1 o 2";
71                     }while(estado_esposa<1 || estado_esposa>2);
72
73                     if(estado_esposa == 1)
74                     {
75                         bono_esposa = 0;
76                     }
77                     else
78                     {
79                         bono_esposa = 200;
80                     }
81                 }break;
82                 case 2 :{
83                     bono_esposa = 0;
84                 }break;
85             }
86         }
87     }
88 }
```

Librerías propias del lenguaje

Librerías creadas por programador

```
83
84
85 do{
86     cout<<"\n\n\tTiene Hijos?\n"<<endl;
87     cout<<"\t1. SI"<<endl;
88     cout<<"\t2. NO"<<endl;
89     cout<<"\n";
90     cout<<"\tIngrese opcion(1 o 2): \t";
91     opc_hijos=leedatofPositivos();
92     if(opc_hijos<1 || opc_hijos>2)
93     {
94         cout<<"\tERROR. vuelva a ingresar, la opcion es 1 o 2";
95     }while(opc_hijos<1 || opc_hijos>2);
96
97     switch (opc_hijos)
98     {
99         case 1 :{ do{
100             cout<<"\tIngrese cantidad de hijos: \t";
101             cant_hijos=leedatofPositivos();
102             if(cant_hijos<0)
103             {
104                 cout<<"\tError, la cantidad.\n";
105             }while(cant_hijos<0);
106             if(cant_hijos <= 2)
107             {
108                 bono_hijo = 100;
109             }
110             else
111             {
112                 bono_hijo = 300;
113             }
114             do{
115                 cout<<"\n\n\tHijos Universidad?\n"<<endl;
116                 cout<<"\t1. SI"<<endl;
117                 cout<<"\t2. NO"<<endl;
118                 cout<<"\n";
119                 cout<<"\tIngrese opcion(1 o 2): \t";
120                 opc_universidad=leedatofPositivos();
121                 if(opc_universidad<1 || opc_universidad>2)
122                 {
123                     cout<<"\tERROR. vuelva a ingresar, la opcion es 1 o 2.\n";
124                 }while(opc_universidad<1 || opc_universidad>2);
125                 if(opc_universidad == 1)
126                 {
127                     do{
128                         cout<<"\tIngrese cantidad de hijos en la universidad: \t";
129                         hijos_universidad=leedatofPositivos();
130                         if(hijos_universidad>cant_hijos)
131                         {
132                             cout<<"\tERROR.Escedio el total de hijos";
133                             cout<<"\n";
134                         }
135                         while (hijos_universidad>cant_hijos);
136                         bono_universidad = hijos_universidad * 100;
137                     }
138                     else
139                     {
140                         bono_universidad = 0;
141                     }
142                 }break;
143                 case 2 :{
144                     bono_universidad=0;
145                 }break;
146             }
147         }
148     }
```

```
148
149     cout<<"\tEl bono esposa es: "<<bono_esposa<<"\n";
150     cout<<"\tEl bono hijo es: \t"<<bono_hijo<<"\n";
151     cout<<"\tEl bono hijo en universidad es: \t"<<bono_universidad<<"\n";
152     sueldo_basico= sueldo + bono_esposa + bono_hijo + bono_universidad;
153     cout<<"\tEl sueldo base es: \t"<<sueldo_basico<<"\n";
154     cout;
155 }
156
157 void Salir()
158 {
159     cout << "\n\tGracias..Vuelva pronto...!\n";
160 }
161
162 int Menu()
163 {
164     char rpt,letra;
165     int op;
166
167     cout << "\n\t* ----- MENU ----- * \n";
168     cout << "\t* 1. Procesar. * \n";
169     cout << "\t* 2. Salir. * \n";
170
171     do { cout << "\tIngrese opcion (1 o 2):\t";
172         op = leedatofPositivos();
173         if(op<1 || op>3)
174             cout << "\tERROR. Vuelva a Ingresar.";
175     } while (op<1 || op>3);
176
177     switch (op)
178     {
179         case 1 :{ proceso();
180             cout << "\n";
181             }break;
182         case 2 :{ do{
183             cout << "\tDesea Salir del programa? (S/N):\t";
184             letra = leedatoc();
185             if (isalpha(letra))
186             {
187                 rpt = toupper(letra);
188                 if (rpt != 'S' && rpt != 'N')
189                     cout << "\tError! Ingrese solo S o N. Vuelva a Ingresar." << "\n\n";
190             }
191             else
192             {
193                 cout << "\tSolo debe ingresar letras." << "\n\n";
194                 rpt = 'T';
195             }
196             while (rpt != 'S' && rpt != 'N');
197             if(toupper(letra) == 'S')
198                 Salir();
199             else
200                 Menu();
201             }break;
202     }
203     return op;
204 }
205
206 int main()
207 {
208     int rpt;
209     system("color 1b");
210
211     do {
212         rpt = Menu();
213     } while (rpt != 2);
214
215     system("pause");
216     return 0;
217 }
```

RECUSIÓN: invocado en la línea 203

RECUSIÓN



Carpeta: Programas Semana 15

Nombre Proyecto: Proyecto2_VendePanLibreriasRecurcion

Nombre Programa: Programa2_VendePanLibreriasRecurcion

Una vendedora de pan, tiene un stock inicial y del cual atiende pedidos a varios clientes.

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

// Declarar variables globales
int inventarioPan, cantPan;
int contClientAtendido = 0, contClienteNoAtendido = 0, TotalVendido = 0;
```

Invoca a Librerías, según de programa anterior.

```
// ---MODULOS DE LECTURA
float leedatof()
{
    float dato;
    cin >> dato;
    return dato;
}

int leedatof()
{
    int dato;
    cin >> dato;
    return dato;
}
```

Ingresar datos validando números y letras, según de programa anterior.

```
// ---- MODULO SALIR
void Salir()
{
    cout << "\n";
    cout << "\t\tGracias..Vuelva pronto..!!.\n";
}

// ---- MODULO Mostrar
void Mostrar()
{
    cout << "\n";
    cout << "\t\tCantidad de atendidos:\t\t" << contClientAtendido << endl;
    cout << "\t\tCantidad de no atendidos:\t\t" << contClienteNoAtendido << endl;
    cout << "\t\tAcumulado de ventas:\t\t" << TotalVendido << endl;
    cout << "\t\tQueda en el inventario:\t\t" << inventarioPan << endl;
    cout << "\n";
}
```

```
// ----- MODULO VENTA
void Venta()
{
    do {
        cout << "\tIngrese cantidad panes por clientes:\t";
        cantPan = leedatof();
        if (cantPan <= 0)
            cout << "\tERROR. Vuelva a ingresar >0." << endl;
    } while (cantPan <= 0);

    if (cantPan <= inventarioPan)
    {
        contClientAtendido = contClientAtendido + 1;
        inventarioPan = inventarioPan - cantPan;
        TotalVendido = TotalVendido + cantPan;
    }
    else
    {
        cout << "\n";
        cout << "\tNo se puede atender esta cantidad." << endl;
        contClienteNoAtendido = contClienteNoAtendido + 1;
    }
}
```

```
// ----- MODULO QUE PROCESA (SUB MENU)
void Procesar()
{
    int opc;

    do {
        cout << "\n";
        cout << "\t\t\t---VENTA PAN---\n";
        cout << "\t\t1.Vender" << endl;
        cout << "\t\t2.Reportar" << endl;
        cout << "\t\t3.Salir" << endl;

        cout << "\t\tIngrese opcion\t\t";
        opc = leedatof();

        if (opc != 1 && opc != 2 && opc != 3)
            cout << "ERROR. Ingrese 1 o 2 o 3 ";
    } while (opc != 1 && opc != 2 && opc != 3);

    switch (opc)
    {
        case 1: { Venta();
                  } break;

        case 2: { Mostrar();
                  } break;

        case 3: { Salir();
                  } break;
    }
    while (opc != 3);
}
```

```
//---MODULO MENU (MENU PRINCIPAL)
int Menu()
{
    int op;

    do {
        cout << "\n\n";
        cout << "\t\t\t---- MENU ----";
        cout << "\n";
        cout << "\t\t1. Procesar." << endl;
        cout << "\t\t2. Salir." << endl;
        cout << "\t\tIngrese opcion (1 o 2):\t";
        op = leedatof();
        if (op < 1 || op > 2)
            cout << "\t\tERROR. Vuelva a Ingresar.";
    } while (op < 1 || op > 2);

    switch (op)
    {
        case 1: { Procesar();
                  cout << "\n\n";
                  } break;

        case 2: { Salir();
                  } break;
    }

    return op;
}
```

Usar la recursión del menú, según de programa anterior.

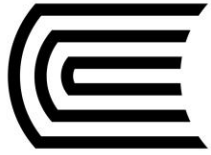
```
//---MODULO PRINCIPAL
int main()
{
    int rpt;

    do {
        cout << "\n";
        cout << "\t\tIngrese Total de panes a vender:\t";
        inventarioPan = leedatof();

        if (inventarioPan <= 0)
            cout << "\t\tERROR.Vuelva ingresar>0" << endl;
    } while (inventarioPan <= 0);

    do {
        rpt = Menu();
    } while (rpt == 1);

    system("pause");
    return 0;
}
```



Conclusiones

Modularización de programas:
Funciones propias del lenguaje de programación y recursividad, con librerías de programación

Se implementan librerías con módulos reutilizables.

En los módulos se puede usar las funciones `isalpha()`, `atoi()`, etc.

El módulo de menú es recursivo por que se invoca a sí mismo.

Programa

```
147 cout<<"\tEl bono esposa es: "<<bono_esposa<<"\n";
148 cout<<"\tEl bono hijo es: \t"<<bono_hijo<<"\n";
149 cout<<"\tEl bono hijo en universidad es: \t"<<bono_universidad<<"\n";
150
151 sueldo_basico= sueldo + bono_esposa + bono_hijo + bono_universidad;
152
153 cout<<"\tEl sueldo base es: \t"<<sueldo_basico<<"\n";
154 cout;
155
156 }
157
158 void Salir()
159 {
160     cout << "\n\t\tGracias..Vuelva pronto...!\n";
161 }
162
163 int Menu() ← RECURSIÓN invocada en la línea 203
164 {
165     char rpta, letra;
166     int op;
167
168     cout << "\n\t* ---- MENU ---- *\n";
169     cout << "\t* 1. Procesar. *\n";
170     cout << "\t* 2. Salir. *\n";
171
172     do {
173         cout << "\tIngrese opcion (1 o 2 o 3):\t";
174         op = leedatocPositivos();
175         if(op<1 || op >3)
176             cout << "\tERROR. Vuelva a Ingresar.";
177     } while (op<1 || op >3);
178
179     switch (op)
180     {
181         case 1: {
182             proceso();
183             cout << "\n";
184             break;
185         }
186         case 2: {
187             do {
188                 cout << "\tDesea Salir del programa? (S/N):\t";
189                 letra = leedatoc();
190                 if (isalpha(letra))
191                     rpta = toupper(letra);
192                 if (rpta != 'S' && rpta != 'N')
193                     cout << "\tError! Ingrese solo S o N. Vuelva a Ingresar." << "\n\n";
194             } else {
195                 cout << "\tSolo debe ingresar letras." << "\n\n";
196                 rpta = 'T';
197             } while (rpta != 'S' && rpta != 'N');
198             if(toupper(letra) == 'S')
199                 Salir();
200             else
201                 Menu(); ← RECURSIÓN
202         }
203     }
204 }
205
206 return op;
207
208 }
```





Propósito

Práctica

Crea programas con librerías de programación con Funciones propias del lenguaje de programación y recursividad, en el lenguaje de programación C/ C++.



Consolidado 2

Evaluación 2

Indicaciones para su desarrollo



Conclusiones

Modularización de programas:
Funciones propias del lenguaje de programación y recursividad, con librerías de programación

Se implementan librerías con módulos reutilizables.

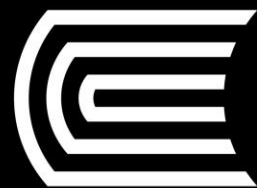
En los módulos se puede usar las funciones `isalpha()`, `atoi()`, etc.

El módulo de menú es recursivo por que se invoca a sí mismo.

Programa

```
147 cout<<"\tEl bono esposa es: "<<bono_esposa<<"\n";
148 cout<<"\tEl bono hijo es: \t"<<bono_hijo<<"\n";
149 cout<<"\tEl bono hijo en universidad es: \t"<<bono_universidad<<"\n";
150
151 sueldo_basico= sueldo + bono_esposa + bono_hijo + bono_universidad;
152
153 cout<<"\tEl sueldo base es: \t"<<sueldo_basico<<"\n";
154 cout;
155
156 }
157
158 void Salir()
159 {
160     cout << "\n\t\tGracias..Vuelva pronto...!\n";
161 }
162
163 int Menu() ← RECURSIÓN invocada en la línea 203
164 {
165     char rpta, letra;
166     int op;
167
168     cout << "\n\t* ---- MENU ---- *\n";
169     cout << "\t* 1. Procesar. *\n";
170     cout << "\t* 2. Salir. *\n";
171
172     do {
173         cout << "\tIngrese opcion (1 o 2 o 3):\t";
174         op = leedatocPositivos();
175         if(op<1 || op >3)
176             cout << "\tERROR. Vuelva a Ingresar.";
177     } while (op<1 || op >3);
178
179     switch (op)
180     {
181         case 1: {
182             proceso();
183             cout << "\n";
184             break;
185         }
186         case 2: {
187             do {
188                 cout << "\tDesea Salir del programa? (S/N):\t";
189                 letra = leedatoc();
190                 if (isalpha(letra))
191                     rpta = toupper(letra);
192                 if (rpta != 'S' && rpta != 'N')
193                     cout << "\tError! Ingrese solo S o N. Vuelva a Ingresar." << "\n\n";
194             } else {
195                 cout << "\tSolo debe ingresar letras." << "\n\n";
196                 rpta = 'T';
197             } while (rpta != 'S' && rpta != 'N');
198             if(toupper(letra) == 'S')
199                 Salir();
200             else
201                 Menu(); ← RECURSIÓN
202         }
203     }
204 }
205
206 return op;
207
208 }
```





ucontinental.edu.pe