

## Guía práctica de laboratorio de Fundamentos de Programación

**Cuarta Unidad:** Módulos para la programación: funciones propias del lenguaje de programación – recursividad

### Práctica N°13

Sección :

Docente :

Apellidos :

Nombres :

Fecha : / / Duración: 100 min

Tipo de práctica: Individual ( x ) Grupal ( )

**Instrucciones:** Desarrolle los siguientes programas haciendo uso del software Dev-C++.

**1.- Tema:** Funciones propias del lenguaje de programación

**2.- Objetivo:** Poner en práctica los principios básicos de librerías en lenguaje C++.

**3.- Materiales a utilizar en la práctica:**

- Programa Dev C++
- Procesador de texto Microsoft Word
- Computadora personal

**4.- Planteamiento de ejercicios:**

---

#### FUNCIONES PROPIAS DEL LENGUAJE DE PROGRAMACIÓN

---

Librería	Función
stdlib.h	abs( )
stdlib.h	labs( )
stdlib.h	atof( )
stdlib.h	atoi( )
stdlib.h	atol( )
stdlib.h	exit( )
stdlib.h	rand( )
stdlib.h	system( )
math.h	ceil( )
math.h	fabs( )
math.h	fmod( )
math.h	pow( )
math.h	sqrt( )
stdio.h	getchar( )
stdio.h	putchar( )
stdio.h	gets( )
stdio.h	puts( )
string.h	strcat( )
string.h	strcmp( )
string.h	strcpy( )

string.h	strcspn( )
string.h	strerror
string.h	strlen( )
ctype.h	tolower( )
ctype.h	toupper( )
type.h	isalnum( )
ctype.h	isalpha( )
ctype.h	isascii( )
ctype.h	isdigit( )
ctype.h	islower( )
ctype.h	isupper( )
ctype.h	ispunct( )
ctype.h	isspace( )
ctype.h	isxdigit( )

### Ejercicio 01:

Elaborar el programa que permita reportar el monto acumulado de ventas y la cantidad de clientes atendidos. Considere diferenciar mayúsculas y minúsculas (toupper), la conversión del ingreso de cadenas a entero (atoi) o a decimal (atof), que permita diferenciar letras de dígitos (isalpha) y que permita ingresar más de un dígito (string Cadenalngresada)

### SOLUCIÓN



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

```

59 { cantart = atoi(CadenaIngresada.c_str()); // usando string...permite convertir con la funcion c_str()
60 if (cantart <= 0)
61     cout << "\t\t ERROR. Tiene que ser un numero positivo.\n";
62 }
63 } while (cantart <= 0);
64
65 if (cantart > 100)
66     dcto = 0.4;
67 else
68     if (cantart >= 25 && cantart <= 100)
69         dcto = 0.2;
70     else
71         if (cantart >= 10 && cantart <= 25)
72             dcto = 0.1;
73         else
74             if (cantart < 10)
75                 { dcto = 0.0;
76                   cout << "\n\n" << " \t\t\t No tiene descuento." << "\n\n";
77                 }
78
79 MontBruto = precio * cantart;
80 montdsct = MontBruto * dcto;
81 montofinal = MontBruto - montdsct;
82
83 cout << "\n\n";
84 cout << "Monto sin descuento es:\t\t\t" << MontBruto;
85 cout << "\n\n";
86 cout << "Monto de descuento es:\t\t\t" << montdsct;
87 cout << "\n\n";
88 cout << "Monto a pagar es:\t\t\t" << montofinal;
89 cout << "\n\n";
90
91 return montofinal;
92 }
93
94 void main()
95 {
96     char rpt;
97     int cont = 0;
98     double Totalvendido = 0;
99     double montofinal;
100
101     do{ montofinal = ventas();
102         cont++; // contador de ventas
103         Totalvendido = Totalvendido + montofinal; // acumulador de ventas
104         rpt = repeticion();
105     } while (rpt == 'S');
106
107     cout << "\n\n";
108     cout << "Acumulador de Venta es:\t\t\t" << Totalvendido;
109     cout << "\n\n";
110     cout << "Contador de Ventas es:\t\t\t" << cont;
111     cout << "\n\n";
112     cout << "\n\n";
113     system("pause");
114 }

```

## Ejercicio 02:

Elaborar el programa que permita ingresar el valor del promedio de una nota, validando que esté entre cero a veinte.



ConvierteDatos.h

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

```

1  #include<iostream>
2  #include "ConvierteDatos.h"
3  #include<string>
4  using namespace std;
5
6  int prom;
7
8  char leedatoc()
9  {   char dato;
10     cin >> dato;
11     return dato;
12 }
13
14 char repeticion()
15 {   char letra;
16     char rpt;
17
18     do
19     {   cout << "Desea continuar? (S/N):\t\t\t";
20         letra = leedatoc();
21         if (isalpha(letra))
22         {   rpt = toupper(letra);
23             if (rpt != 'S' && rpt != 'N')
24                 cout << "Error! Ingrese solo S o N. Vuelva a Ingresar." << "\n\n";
25         }
26         else
27         {   cout << "Solo debe ingresar letras." << "\n\n";
28             rpt = 'T';
29         }
30     } while (rpt != 'S' && rpt != 'N');
31     cout << "\n";
32     return rpt;
33 }
34
35 void main()
36 {   char rpt;
37
38     do{
39         do{
40             prom = leedatoepositivos();
41             if (prom <0 || prom >20)
42                 cout << "\t\t ERROR.Ingrese de 0 a 20.\n";
43         } while (prom <0 || prom >20); // con prom = -5 valor cualquiera
44
45         cout << "\t\n";
46         rpt = repeticion();
47     }while (rpt == 'S');
48
49     cout << "\n\n";
50     system("pause");
51 }
52

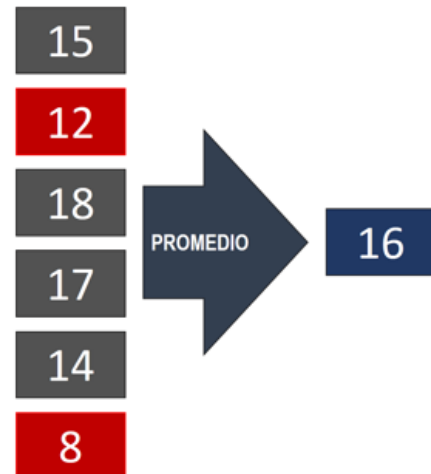
```

### Ejercicio 03

Desarrollar un programa para encontrar la moda de un arreglo.

#### Ejercicio 04

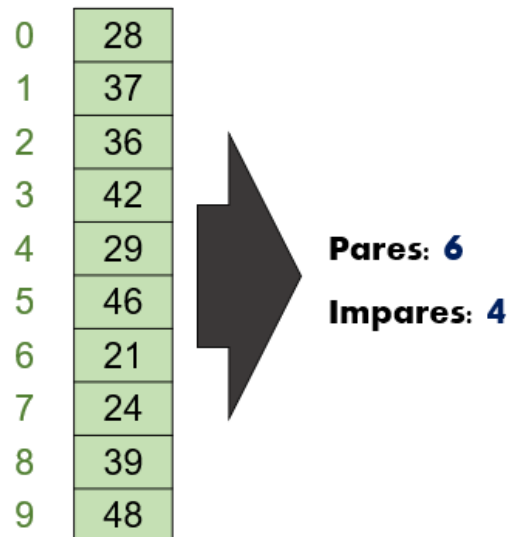
La evaluación de un estudiante se basa en 6 notas, de las cuales se consideran solo las 4 mejores notas.



#### Ejercicio 05

Desarrollar un programa que genere un arreglo de 10 números enteros entre 20 y 50.

Luego debe contar cuantos números son pares y cuantos son impares.



#### Ejercicio 06

Desarrollar un programa que genere una matriz de 4x3 de números aleatorios comprendidos entre 10 y 30.

Luego debe calcular la suma de cada columna.

Finalmente, el programa debe mostrar la matriz y la suma de sus columnas.

	0	1	2
0	15	19	29
1	22	18	13
2	25	25	25
3	17	28	18
Suma	79	90	85

### Ejercicio 07

Desarrollar un programa que permita a un profesor ingresar la nota de sus alumnos, se debe validar que la nota se encuentre entre 0 y 20.

Luego el programa debe mostrar los siguientes datos estadísticos:

- Cantidad de aprobados
- Cantidad de desaprobados
- La nota mayor
- La nota menor
- La nota promedio

#### REPORTE

- Cantidad de alumnos: 10
- Aprobados: 6
- Desaprobado: 4
- Nota mayor: 18
- Nota menor: 10
- Nota promedio: 15

### Ejercicio 08

En una empresa trabajan 5 personas, cada una tiene roles diferentes por lo tanto la cantidad de horas que trabajan a la semana es diferente y su pago por hora también lo es.

Hacer un programa que permita el ingreso de los datos de cada trabajador para poder calcular su pago y el pago total que debe hacer la empresa.

NOMBRE	HORAS	PAGO x HORA	IMPORTE
PEDRO	20	30	600
KARLA	40	20	800
GUSTAVO	40	60	2400
DANIELA	30	40	1200
LEONOR	35	30	1050

**PAGO TOTAL: 6,050.00 Dólares**

### 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 Tábora, A. (2008). Introducción a la computación. México: Pearson Educación.