



## Tarea: Arreglos unidimensionales y funciones en C++

Objetivo general:

Aplicar el uso de arreglos unidimensionales junto con diferentes tipos de funciones (void, por referencia, por valor, y con retorno) y aplicar un método de ordenamiento (Burbuja) utilizando C++.

### Funciones con void (sin retorno)

Enunciado 1: Diseñe un programa en C++ que permita ingresar cinco números enteros en un arreglo unidimensional y luego, mediante una función void, muestre los valores almacenados en pantalla.

```
ejerc1.cpp X  ejer2.cpp  FUNCIONES.cpp  Code algoritmos X
algoritmos >  ejerc1.cpp >  mostrar(int [5])
1  #include<iostream>
2  using namespace std;
3  //funcion para mostrar datos
4  void mostrar(int lista[5]){
5      cout<<"valores ordenados \n";
6      for (int i=0;i<5;i++){
7          cout<<lista[i]<<endl;
8      }
9  }
10 int main(){
11     //MASAPANTA ALEXANDER
12     int valor;
13     cout<<"ingrese 5 valores \n ";
14     int lista[5];
15     for (int i=0;i<5;i++){
16         cin>>lista[i];
17     }
18     //se usa dos for , porq neceisto q se reste primero 0 y luego 1 ,
19     //de lo contrario la lista busca un valor inexistente , lo q causa error
20     for (int j=0;j<5;j++){
21         for (int i=0;i<5-j-1;i++){
22             if (lista[i]>lista[i+1]){
23                 int temp=lista[i];
24                 lista[i]=lista[i+1];
25                 lista[i+1]=temp;
26             }
27         }
28     }
29     mostrar(lista);
30     return 0;
31 }
32
```

Enunciado 2: Cree un programa que solicite al usuario ingresar las calificaciones de 4 estudiantes, las almacene en un arreglo y utilice una función void para mostrar todas las notas



# ESCUELA POLITÉCNICA NACIONAL ESCUELA DE FORMACIÓN DE TECNÓLOGOS DESARROLLO DE SOFTWARE



registradas.

```
ejerc1.cpp X  ejer2.cpp  ejer3.cpp  FUNCIONES.cpp  Code algoritmos X
algoritmos > ejerc1.cpp > ...
1  #include<iostream>
2  #include<cstdio>
3  using namespace std;
4  //funcion para mostrar datos
5  void mostrar(int* lista,int valor){
6      //especifico q estoy usando lista dinamica
7      cout<<"NOTAS REGISTRADAS \n";
8      for (int i=0;i<4;i++){
9          cout<<lista[i]<<endl;
10     }
11 }
12 int* list(int& valor){
13     //hice una funcion para mejorar los tiempos de los siguientes ejercicio
14     //uso listas dinamicas para permitir al usuario ingresar el valor.
15     cout<<"cuantos numeros va a ingresar? = ";
16     cin>>valor;
17     int* lista=new int[valor];
18     printf("ingrese las %d calificaciones \n",valor);
19     for (int i=0;i<valor;i++){
20         cin>>lista[i];
21     }
22     return lista;
23 }
24 int main(){
25     //MASAPANTA ALEXANDER
26     int valor;
27     int* lista=list(valor);
28     mostrar(lista,valor);
29     delete[] lista;
30     return 0;
31 }
```

```
413x@MacBook-Pro-de-Alexander algoritmos/++/algoritmos/" && g++ ejerc1.cpp -o e
C++/algoritmos/"ejerc1
cuantos numeros va a ingresar? = 4
ingrese las 4 calificaciones
10
9
8
7
NOTAS REGISTRADAS
10
9
8
7
413x@MacBook-Pro-de-Alexander algoritmos/++/algoritmos/" && g++ ejer3.cpp -o e
+/algoritmos/"ejer3
10
413x@MacBook-Pro-de-Alexander algoritmos/++/algoritmos/" && g++ ejer3.cpp -o e
+/algoritmos/"ejer3
15
413x@MacBook-Pro-de-Alexander algoritmos/++/algoritmos/" && g++ ejer3.cpp -o e
+/algoritmos/"ejer3
15
```



## Funciones con parámetros por referencia

Enunciado 3: Desarrolle un programa que lea los elementos de un arreglo de 5 números enteros y, mediante una función que use paso por referencia, calcule la suma total de los elementos.

```
ejerc1.cpp  ejer2.cpp  ejer3.cpp x  ejer4.cpp  FUNCIO  Code algoritmos x
algoritmos > ejer3.cpp > suma(int [], int &)
1  #include<iostream>
2  using namespace std;
3  //funcion para mostrar datos(misma del anterior ejercicio)
4  void mostrar(int lista[],int valor){
5      //especifico la lista
6      cout<<"numeros en lista \n";
7      for (int i=0;i<valor;i++){
8          cout<<lista[i]<<endl;
9      }
10 //funcion para mostrar datos
11 void suma(int valor,int lista[],int& total){
12     total=0;
13     cout<<"el total es = ";
14     for (int i=0;i<valor;i++){
15         total+=lista[i];
16     }
17 }
18 int main(){
19     //MASAPANTA ALEXANDER
20     int valor,total;
21     valor=5;
22     int lista[5]={1,2,3,4,5};
23     mostrar(lista,valor);
24     suma(valor,lista,total);
25     cout<<total;
26     return 0;
27 }
```

```
4l3x@MacBook-Pro-de-A
++/algoritmos/" && g+
+/algoritmos/"ejer3
numeros en lista
1
2
3
4
5
el total es = 15
4l3x@MacBook-Pro-de-A
```

Enunciado 4: Implemente un programa que permita almacenar varios números en un arreglo e identifique, mediante una función que use parámetros por referencia, cuál es el valor mayor del



# ESCUELA POLITÉCNICA NACIONAL ESCUELA DE FORMACIÓN DE TECNÓLOGOS DESARROLLO DE SOFTWARE



arreglo.

```
algoritmos > ejer4.cpp > main()
1 #include <iostream>
2 using namespace std;
3 //creo un puntero que guarde los valores
4 //no uso biblioteca vector para ver otros metodos
5 int* lista= nullptr;
6 //limite inicial 0 , va a crecer conforme el usuario ingrese datos
7 int limite=0;
8 //funcion para mostrar datos(misma del anterior ejercicio)
9 void mostrar(){
10     cout<<"numeros en lista \n";
11     for (int i=0;i<limite;i++){
12         cout<<lista[i]<<" ";
13     }
14     string minuscula(string texto){
15         //funcion q retorna letras en minusculas,c con referencia
16         //cambia cada letra del texto por minusculas ,y retorna la nueva palabra
17         for (char &c:texto){
18             c=tolower(c); }return texto;}
19 void list(string entrada){
20     try{//intenta convertir la entrada a entero
21         //si falla sale entrada invalida
22         int valor=stoi(entrada);
23         //creo un puntero que pueda crecer , inicialmente es 1,
24         int* nuevo= new int[limite+1];
25         //hago que todo lo nuevo se copie en lista, de esta manera aumento el limite de lista
26         nuevo[i]=lista[i];
27         //en lista nueva(NUEVO),de indice inicial 0 , se guardara el valor ingresado
28         nuevo[limite]=valor;
29         //borro los elementos de la lista,(ahorro de memoria)
30         delete[] lista;
31         //ahora vuelvo a copiar los elementos en una lista con mas espacio.
32         lista=nuevo;
33         //aumento el limite tanto de valor como de lista
34         limite++; catch(...){
35             cout<<"Entrada no valida\n";}}
36 int main(){
37     //MASAPANTA ALEXANDER
38     int valor,total;
39     string entrada;
40     int lista;
41     while (true){
42         cout<<"Ingresar valores (escriba salir para terminar)\n";
43         cin>>entrada;
44         if (minuscula(entrada)=="salir"){
45             break;}
46         else{
47             list(entrada);}}
48     mostrar();
49     return 0;}
```

## Funciones con parámetros por valor

Enunciado 5: Escriba un programa que almacene varios números en un arreglo y, mediante una función con parámetros por valor, multiplique cada elemento por un número ingresado por el



usuario, mostrando el resultado en pantalla sin alterar el arreglo original.

```
algoritmos > G+ ejer5.cpp > main()
1  #include<iostream>
2  using namespace std;
3  //declaro un limite global
4  int limite=10;
5  void mostrar(int lista[]){
6  //muestra los valores, en funcion de lista[]
7      cout<<"numeros en lista \n";
8      for (int i=0;i<limite;i++){
9          cout<<lista[i]<<endl;}}
10 void mult(int valor,int lista[]){
11 //creo una copia para no afectar la lista original
12     int copia[limite];
13     for (int i=0;i<limite;i++){
14         copia[i]=lista[i]*valor;
15     }
16 //uso la funcion ante declarada para mostrar la copia
17     mostrar(copia);
18 }
19 int main(){
20     int valor;
21     int lista[10]={};
22     for (int i=0;i<20;i++){
23         lista[i]=i+2;
24     }
25     mostrar(lista);
26     cout<<"Ingrese valor para multiplicar = ";
27     cin>>valor;
28     cout<<"RESULTADO DE ";
29     //aquí ya se realiza el proceso y se muestra
30     mult(valor,lista);
31     //imprimo la lista original para mostrar que no cambia
32     mostrar(lista);
33 }
```

```
413x@MacBook-Pro-de-Alexar
++/algoritmos/" && g++ eje
++algoritmos/"ejer5
numeros en lista
2
3
4
5
6
7
8
9
10
11
Ingrese valor para multip
RESULTADO DE numeros en l
4
6
8
10
12
14
16
18
20
22
numeros en lista
2
3
4
5
6
7
8
9
10
11
zsh: abort      "/Users/41
413x@MacBook-Pro-de-Alexar
```

Enunciado 6: Cree un programa que permita almacenar números enteros en un arreglo e implemente una función con paso por valor que cuente cuántos de esos números son pares. La



función debe retornar la cantidad encontrada.

```
algoritmos > G+ ejerc6.cpp > main()
1  #include<iostream>
2  using namespace std;
3  void mostrar(int lista[]){
4  //muestra los valores, en funcion de lista[]
5      cout<<"numeros en lista \n";
6      for (int i=0;i<20;i++){
7          cout<<lista[i]<<endl;}}
8  void parin(int lista[],int& par,int& impar){
9  //declaro valores , uno q reciba lista , y los demas para guardar
10 //la cantidad, uso referencia para que modifiquen el valor
11 //Alexander Masapanta
12     par=0;
13     impar=0;
14     for (int i=0;i<20;i++){
15         if (lista[i]%2==0){
16             par+=1;
17         }
18         else{
19             impar+=1;
20         }
21     }
22 }
23 int main(){
24 //las defino dentro de main para que guarden valores , una vez pasen por la
25 //funcion parin.
26     int par,impar;
27     int valor;
28     int lista[20]={};
29     //me ahorro escribir datos
30     for (int i=0;i<20;i++){
31         lista[i]=i*(i-1);
32     }
33     mostrar(lista);
34 //aqui guardan los valores , y luego los muestro
35     parin(lista,par,impar);
36     cout<<"los pares encontrados fueron="<<par<<endl;
37     cout<<"los pares encontrados fueron="<<impar<<endl;
38     return 0;
39 }
```

```
Code algoritmos X
35
48
63
80
99
120
143
168
195
224
255
288
323
360
los pares encontrados fueron=10
los pares encontrados fueron=10
413x@MacBook-Pro-de-Alexander algoritmos %
++/algoritmos/" && g++ ejerc6.cpp -o ejerc6
C++/algoritmos/"ejerc6
numeros en lista
0
0
2
6
12
20
30
42
56
72
90
110
132
156
182
210
240
272
306
342
los pares encontrados fueron=20
los pares encontrados fueron=0
413x@MacBook-Pro-de-Alexander algoritmos %
```

## Ordenamiento de arreglos unidimensionales (Método Burbuja)

Enunciado 7: Elabore un programa que permita ordenar un arreglo de números enteros de menor a mayor utilizando el método burbuja. Debe mostrar el arreglo antes y después del



ordenamiento.

```
algoritmos > ejer7.cpp > main()
1  #include<iostream>
2  using namespace std;
3  //la denomino global para poder experimentar con la cantidad de datos
4  int n=10;
5  void mostrar(int lista[]){
6  //muestra los valores, en funcion de lista[]
7      cout<<"Numeros en lista \n";
8      for (int i=0;i<n;i++){
9          cout<<lista[i]<<endl;}}
10 void burbuja(int lista[]){
11     for (int i=0;i<n;i++){
12         //le resto uno , ademas de i a n , para que no busque una posicion inexistente
13         //q seria la 11 , dado que vota error
14         //esa es la razon por la q ademas ocupo dos for
15         for (int j=0;j<n-i-1;j++){
16             if (lista[j]>lista[j+1]){
17                 int after=lista[j];
18                 lista[j]=lista[j+1];
19                 lista[j+1]=after;
20             }
21         }
22     }
23     cout<<"LISTA DE MENOR A MAYOR\n";
24     mostrar(lista);
25 }
26 int main(){
27     int lista[10]={6,4,7,8,9,4,0,3,1,2};
28     mostrar(lista);
29     burbuja(lista);
30     //Alexander Masapanta
31     return 0;
32 }
```

```
413x@MacBook-Pro-de-Alexander
++algoritmos/" && g++ ejer7
C++/algoritmos/"ejerc7
Numeros en lista
6
4
7
8
9
4
0
3
1
2
LISTA DE MENOR A MAYOR
Numeros en lista
0
1
2
3
4
4
6
7
8
9
413x@MacBook-Pro-de-Alexander
```

### Ejercicio con return (función que devuelve un valor)

Enunciado 8: Realice un programa que calcule y retorne el promedio de los elementos de un arreglo (función con return). La función deberá recibir el arreglo y su tamaño, calcular el promedio y devolverlo como double. En el programa principal se mostrará el promedio con dos





decimales.

```
ejerc6.cpp  ejerc7.cpp  ejerc8.cpp x  FUNCIONES.cpp  Calibri  Code algoritmos x
algoritmos > ejerc8.cpp > main()
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4  double prom(int lista[],int size){
5      int total=0;
6      for (int i=0;i<size;i++){
7          total+=lista[i];
8      }
9      double promedio=total/size;
10     return promedio;
11 }
12 int main(){
13     int lista[10]={2,5,4,3,6,8,5,4,3};
14     double promedio=prom(lista,10);
15     cout<<"el promedio es = "<<fixed<<setprecision(2)<<promedio;
16     return 0;
17 }
```

### Factura con descuentos (condicionales)

Enunciado 9: Desarrolle un programa en C++ que lea: nombre del producto, precio unitario y cantidad. Aplique descuentos según la cantidad:

Si la cantidad es **3 o más**, aplicar **15%** de descuento sobre subtotal.

Además, si el subtotal (antes de descuentos) supera 200, aplicar un **descuento adicional de 5%** (acumulativo).

Implemente funciones para: cargar datos (void), calcular subtotal y descuentos (double con return), aplicar descuentos por referencia (modificando el total), y mostrar la factura





(void). Mostrar subtotal, descuentos aplicados y total final.

```
algoritmos > C- ejer9.cpp > main()
36 vector<double> calculo(double precio[],int cant[],double& pago){
46 //guardo el descuento y que hacer en cada caso
47 double descuento (double precio[],int cantidad[],string& mensaje,double& total,double& pago){
48 if (limite>=3 and pago>200 ){
49 //guardo un mensaje personalizado , de acuerdo a cada caso .
50 mensaje="Lleva mas de 2 objetos distintos\nEl subtotal es mayor a 200 \ndescuento total de 20%";
51 total+=pago*(0.80);
52 else if(limite==3){
53 mensaje="Lleva mas de 2 objetos distintos\nAplicada a un descuento del 15%";
54 total+=pago*(0.85);
55 }
56 else{
57 if (pago>200){
58 mensaje="El subtotal es mayor a 200 \ndescuento total de 5%";
59 total+=pago*(0.95);
60 }
61 else{
62 total=pago;
63 }
64 }return total;
65 }
66 void mostrar(vector<double> subtotal,string mensaje,string order [],double total){
67 for (int i=0;i<limite;i++){
68 cout<<i+1<<" ";<<order[i]<<" = $"<<subtotal[i]<<endl;
69 }
70 cout<<mensaje<<"\nTotal a pagar="<<total;
71 }
72 string minuscula(string texto){
73 //codigo para transformar cualquier texto a minuscula
74 for (char &c:texto){
75 c=tolower(c);
76 }
77 return texto;
78 }
79 int main(){
80 //indico punteros que no apunten a nada , todavia .
81 double* precio=NULLptr;
82 string* order=NULLptr;
83 int* cantidad=NULLptr;
84 //aquí defino cada variable para poder guardarles valores , al pasarlos por referencia
85 string entrada,mensaje;
86 double price;
87 int cant;
88 double total=0;
89 double pago=0;
90 //uso while para que pida datos hasta que escriba salir
91 while (true){
92 cout<<"EL TUTI\n";
93 }
```

## Tarjeta Metro/Bus Quito (recarga y pagos)

Enunciado 10: Crea un programa en C++ que simule una **tarjeta de transporte del Metro/Bus de Quito**.

El programa debe permitir **recargar saldo, pagar pasajes y mostrar el saldo disponible después de cada pasada**.

La tarjeta debe tener un **saldo inicial de 0**.

Debe permitir **recargar** un monto (por ejemplo, \$5, \$10, etc.).

Cada vez que el usuario pague un pasaje:

- Se **descuenta** el valor del pasaje (por ejemplo, \$0.45).
- Se **muestra el saldo actual** después del pago.

El programa debe llevar un **contador de pasadas** (viajes pagados).

Usar **funciones** para organizar el código:



ESCUELA POLITÉCNICA NACIONAL  
ESCUELA DE FORMACIÓN DE TECNÓLOGOS  
DESARROLLO DE SOFTWARE



- void recargar(double &saldo, double monto)
- bool pagar(double &saldo, double tarifa)
- void mostrarEstado(double saldo, int pasadas)

===== TARJETA METRO/BUS QUITO =====

Saldo actual: \$0.00

Ingrese monto a recargar: 5

Recarga exitosa. Saldo actual: \$5.00

Pagando pasaje de \$0.45 ...

Pago exitoso. Saldo actual: \$4.55

Pagando pasaje de \$0.45 ...

Pago exitoso. Saldo actual: \$4.10

Pasadas realizadas: 2

Saldo final: \$4.10



```
=====
arc7.cpp  G- ejerc8.cpp  G- ejer4.cpp  G- ejer9.cpp  G- ejer10.cpp X  G- FUNCIONE  G- Code algoritmos X
1 // MASAPANTA ALEXANDER
2 #include <iostream>
3 using namespace std;
4
5 // Recargar saldo uso referencia porq debe cambiar la variable
6 void recargar(double &saldo, double monto){
7     saldo += monto;
8     cout << "Recarga exitosa. Saldo actual: $" << saldo << endl;
9 }
10
11 // esta definido como booleano , por esa razon me regresa true o false
12 bool pagar(double &saldo, double tarifa){
13     if (saldo >= tarifa){
14         saldo -= tarifa;
15         cout << "Pago exitoso. Saldo actual: $" << saldo << endl;
16         return true;
17     }
18     else {
19         cout << "Saldo insuficiente. Recargue para continuar.\n";
20         return false;
21     }
22 }
23
24 // Mostrar estado final
25 void mostrarEstado(double saldo, int pasadas){
26     cout << "\n--- ESTADO DE LA TARJETA ---\n";
27     cout << "Pasadas realizadas: " << pasadas << endl;
28     cout << "Saldo actual: $" << saldo << endl;
29 }
30
31 int main(){
32     //uso una tarifa justa .
33     double saldo = 0;
34     double tarifa = 0.45;
35     int pasadas = 0;
36     int opcion;
37
38     do {
39         cout << "\n===== TARJETA METRO/BUS QUITO =====\n";
40         cout << "1. Recargar saldo\n";
41         cout << "2. Pagar pasaje\n";
42         cout << "3. Mostrar estado\n";
43         cout << "4. Salir\n";
44         cout << "Seleccione una opcion: ";
45
46         int opcion;
47         while (opcion < 1 || opcion > 4) {
48             cout << "Opcion invalida. Seleccione una opcion entre 1 y 4: ";
49             opcion = 0;
50         }
51
52         switch (opcion) {
53             case 1:
54                 recargar(saldo, 7);
55                 break;
56             case 2:
57                 pagar(saldo, tarifa);
58                 break;
59             case 3:
60                 mostrarEstado(saldo, pasadas);
61                 break;
62             case 4:
63                 break;
64         }
65     } while (opcion != 4);
66
67     mostrarEstado(saldo, pasadas);
68     cout << "\nSaliendo\n";
69 }
```

## Lineamientos de entrega

1. El trabajo debe desarrollarse en C++ utilizando funciones para cada ejercicio.
2. Cada programa debe incluir:
  - Nombre del estudiante.
  - Descripción breve del ejercicio.
  - Comentarios dentro del código.
3. En github crear un **repositorio** y suba los archivos .cpp y el documento .docx con los códigos correctamente indentados.
4. Poner el enlace de github en un pdf.