



UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE
ÁREA DE CIENCIAS DE LA COMPUTACION
ESTRUCTURA DE DATOS

NRC: 20723
Fecha: June 13, 2025

Autor: Mateo Camacho, Jonathan Diaz, Andy Conteron

Ejercicio: Ordenamiento ASCII de nombres con MVC (clases separadas)

Objetivo: Implementar el patrón MVC con clases especializadas para ordenar nombres por valores ASCII.

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 const int maxNombres = 5;
6 const int maxLongitud = 30;
7
8 // --- Clase Traductor: separar y transcribir ---
9 class Traductor {
10 public:
11     void separar(const char* nombre, int* ascii, int& longitud) {
12         longitud = strlen(nombre);
13         for (int i = 0; i < longitud; ++i) {
14             ascii[i] = nombre[i];
15         }
16     }
17
18     void transcribir(int* ascii, char* nombre, int longitud) {
19         for (int i = 0; i < longitud; ++i) {
20             nombre[i] = static_cast<char>(ascii[i]);
21         }
22         nombre[longitud] = '\0';
23     }
24 };
25
26 // --- Clase BumbbleSort: ordenamiento ---
27 class BumbbleSort {
28 public:
29     void ordenar(int* arreglo, int longitud) {
30         for (int i = 0; i < longitud - 1; ++i) {
```

```

31         for (int j = 0; j < longitud - i - 1; ++j) {
32             if (arreglo[j] > arreglo[j + 1]) {
33                 int temp = arreglo[j];
34                 arreglo[j] = arreglo[j + 1];
35                 arreglo[j + 1] = temp;
36             }
37         }
38     }
39 }
40 };
41
42 // --- Modelo ---
43 class NombreModelo {
44 private:
45     char** nombres;
46 public:
47     NombreModelo() {
48         nombres = new char*[maxNombres];
49         for (int i = 0; i < maxNombres; ++i) {
50             nombres[i] = new char[maxLongitud];
51             nombres[i][0] = '\0';
52         }
53     }
54     ~NombreModelo() {
55         for (int i = 0; i < maxNombres; ++i) {
56             delete[] nombres[i];
57         }
58         delete[] nombres;
59     }
60     char* getNombre(int i) { return nombres[i]; }
61     void setNombre(int i, const char* nombre) {
62         strncpy(nombres[i], nombre, maxLongitud - 1);
63         nombres[i][maxLongitud - 1] = '\0';
64     }
65     char** getNombres() { return nombres; }
66 };
67
68 // --- Vista ---
69 class NombreVista {
70 public:
71     void mostrarMensaje(const char* mensaje) {
72         cout << mensaje;
73     }
74     void mostrarNombresOrdenados(char** nombres, int cantidad) {
75         cout << "\nNombres_ordenados_por_letras_(ASCII):" << endl;
76         for (int i = 0; i < cantidad; ++i) {
77             cout << nombres[i] << endl;
78         }
79     }
80 };
81

```

```

82 // --- Controlador ---
83 class NombreControlador {
84 private:
85     NombreModelo modelo;
86     NombreVista vista;
87     Traductor traductor;
88     BumbbleSort orden;
89
90 public:
91     void iniciar() {
92         // Entrada nombres
93         for (int i = 0; i < maxNombres; ++i) {
94             vista.mostrarMensaje(("Ingrese_nombre_" + to_string(i +
95                                     1) + ":_").c_str());
96             char buffer[maxLongitud];
97             cin.getline(buffer, maxLongitud);
98             modelo.setNombre(i, buffer);
99
100            // Ordenar ASCII de cada nombre
101            for (int i = 0; i < maxNombres; ++i) {
102                int longitud = 0;
103                int* ascii = new int[maxLongitud];
104
105                traductor.separar(modelo.getNombre(i), ascii, longitud)
106                ;
107                orden.ordenar(ascii, longitud);
108                traductor.transcribir(ascii, modelo.getNombre(i),
109                    longitud);
110
111                delete[] ascii;
112            }
113
114            // Mostrar resultado
115            vista.mostrarNombresOrdenados(modelo.getNombres(),
116                maxNombres);
117        }
118    };
119
120 // --- main ---
121 int main() {
122     NombreControlador app;
123     app.iniciar();
124     return 0;
125 }

```

Listing 1: Código C++ con Traductor, BumbbleSort, Controlador y Vista